

REPORT DOCUMENTATION PAGE			1 Form Approved OMB NO. 0704-0188		
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) 29-08-2014		2. REPORT TYPE MS Thesis		3. DATES COVERED (From - To) -	
4. TITLE AND SUBTITLE VERIFICATION AND VALIDATION OF THE SPALART-ALLMARAS TURBULENCE MODEL FOR STRAND GRIDS			5a. CONTRACT NUMBER W911NF-12-1-0008		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER 622307		
6. AUTHORS Oisin Tong			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAMES AND ADDRESSES Utah State University 1415 Old Main Hill - Room 64 Logan, UT 84322 -1415			8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211			10. SPONSOR/MONITOR'S ACRONYM(S) ARO		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S) 60390-EG.40		
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.					
14. ABSTRACT The strand-Cartesian grid approach provides many advantages for complex moving-body flow simulations, including fully-automatic volume grid generation, highly scalable domain connectivity, and high-order accuracy. In this work the Spalart-Allmaras model is implemented, verified, and validated for high Reynolds number turbulent flows in a strand-Cartesian solver. Second-order convergence is achieved using the Method of Manufactured Solutions implying correct implementation of the turbulence model. By using the NASA-Langley online resource, specific flow cases are validated with two independent compressible codes: FUN3D and CFL3D. The strand solver					
15. SUBJECT TERMS Spalart-Allmaras model, strand grids					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Aaron Katz
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU			19b. TELEPHONE NUMBER 435-797-7021

Report Title

VERIFICATION AND VALIDATION OF THE SPALART-ALLMARAS TURBULENCE MODEL FOR STRAND GRIDS

ABSTRACT

The strand-Cartesian grid approach provides many advantages for complex moving-body flow simulations, including fully-automatic volume grid generation, highly scalable domain connectivity, and high-order accuracy. In this work the Spalart-Allmaras model is implemented, verified, and validated for high Reynolds number turbulent flows in a strand-Cartesian solver. Second-order convergence is achieved using the Method of Manufactured Solutions implying correct implementation of the turbulence model. By using the NASA-Langley online resource, specific flow cases are validated with two independent compressible codes: FUN3D and CFL3D. The strand solver is validated with zero-pressure gradient flat plate and bump-in-channel cases, and shows excellent agreement with FUN3D and CFL3D for various aspects of turbulent flow, including: velocity profiles, turbulent viscosity profile, coefficient of surface pressure, and drag. Methods of handling sharp corners with strand grids through combinations of strand vector smoothing, multiple strands emanating from a single surface node, and telescoping Cartesian refinement into corner regions of the near-body grid are investigated for a NACA 0012 case. For standard viscous high-aspect ratio grids, smoothed strands with telescoping Cartesian refinement provide the most accurate results with the least complexity. Mesh discontinuities associated with use of multiple strands at sharp corners produce more error than with smoothed strands. With both strand approaches – vector smoothing and multiple strands – targeted Cartesian refinement is critical to capture features near sharp corners where strand grids alone are too coarse to capture. Other results show agreement with FUN3D and CFL3D. By using strand vector smoothing and telescoping Cartesian refinement, a NACA 4412 trailing edge separation case is validated with comparison against CFL3D and FUN3D. Velocity profiles show reasonable agreement with CFL3D; however implementing preconditioning to the solver in the future may increase the accuracy of the solution.

VERIFICATION AND VALIDATION OF THE SPALART-ALLMARAS
TURBULENCE MODEL FOR STRAND GRIDS

by

Oisin Tong

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Mechanical Engineering

Approved:

Dr. Aaron Katz
Major Professor

Dr. Robert Spall
Committee Member

Dr. Hang Ban
Committee Member

Dr. Mark McLellan
Vice President for Research and
Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2013

Copyright © Oisin Tong 2013

All Rights Reserved

Abstract

Verification and Validation of the Spalart-Allmaras Turbulence Model for Strand Grids

by

Oisin Tong, Master of Science

Utah State University, 2013

Major Professor: Dr. Aaron Katz

Department: Mechanical and Aerospace Engineering

The strand-Cartesian grid approach provides many advantages for complex moving-body flow simulations, including fully-automatic volume grid generation, highly scalable domain connectivity, and high-order accuracy. In this work the Spalart-Allmaras model is implemented, verified, and validated for high Reynolds number turbulent flows in a strand-Cartesian solver. Second-order convergence is achieved using the Method of Manufactured Solutions implying correct implementation of the turbulence model. By using the NASA-Langley online resource, specific flow cases are validated with two independent compressible codes: FUN3D and CFL3D. The strand solver is validated with zero-pressure gradient flat plate and bump-in-channel cases, and shows excellent agreement with FUN3D and CFL3D for various aspects of turbulent flow, including: velocity profiles, turbulent viscosity profile, coefficient of surface pressure, and drag. Methods of handling sharp corners with strand grids through combinations of strand vector smoothing, multiple strands emanating from a single surface node, and telescoping Cartesian refinement into corner regions of the near-body grid are investigated for a NACA 0012 case. For standard viscous high-aspect ratio grids, smoothed strands with telescoping Cartesian refinement provide the most accurate results with the least complexity. Mesh discontinuities associated with use of multiple strands at sharp corners produce more error than with smoothed strands. With both

strand approaches – vector smoothing and multiple strands – targeted Cartesian refinement is critical to capture features near sharp corners where strand grids alone are too coarse to capture. Other results show agreement with FUN3D and CFL3D. By using strand vector smoothing and telescoping Cartesian refinement, a NACA 4412 trailing edge separation case is validated with comparison against CFL3D and FUN3D. Velocity profiles show reasonable agreement with CFL3D; however implementing preconditioning to the solver in the future may increase the accuracy of the solution.

(78 pages)

Public Abstract

Verification and Validation of the Spalart-Allmaras Turbulence Model for Strand Grids

The strand-Cartesian grid approach is a unique method of generating and computing fluid dynamic simulations. The strand-Cartesian approach provides highly desirable qualities of fully-automatic grid generation and high-order accuracy. This thesis focuses on the implementation of the Spalart-Allmaras turbulence model to the strand-Cartesian grid framework. Verification and validation is required to ensure correct implementation of the turbulence model.

Mathematical code verification is used to ensure correct implementation of new algorithms within the code framework. The Spalart-Allmaras model is verified with the Method of Manufactured Solutions (MMS). MMS shows second-order convergence, which implies that the new algorithms are correctly implemented.

Validation of the strand-Cartesian solver is completed by simulating certain cases for comparison against the results of two independent compressible codes; CFL3D and FUN3D. The NASA-Langley turbulence resource provided the inputs and conditions required to run the cases, as well as the case results for these two codes. The strand solver showed excellent agreement with both NASA resource codes for a zero-pressure gradient flat plate and bump-in-channel. The treatment of the sharp corner on a NACA 0012 airfoil is investigated, resulting in an optimal external sharp corner configuration of strand vector smoothing with a base Cartesian grid and telescoping Cartesian refinement around the trailing edge. Results from the case agree well with those from CFL3D and FUN3D. Additionally, a NACA 4412 airfoil case is examined, and shows good agreement with CFL3D and FUN3D, resulting in validation for this case.

Oisin Tong, Master of Science

Acknowledgments

The current work was performed with the support of the Department of Defense High Performance Computing Modernization Office (HPCMO) Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program. Additional support was provided by the Army Research Office Fluid Dynamics Program directed by Dr. Frederick Ferguson.

This work could not have been completed without the support and time of Dr. Aaron Katz.

Oisin Tong

Contents

	Page
Abstract	iii
Public Abstract	v
Acknowledgments	vi
List of Tables	ix
List of Figures	x
1 Introduction	1
1.1 Objectives	3
1.2 Outline of Thesis	4
2 Literary Review	5
2.1 Strand Grids	5
2.2 Turbulence	7
2.3 Turbulence Modeling and One-Equation Turbulence Models	10
2.3.1 Turbulence Modeling	10
2.3.2 One-Equation Turbulence Models	11
2.4 Verification and Validation	12
2.4.1 Verification	12
2.4.2 Validation	13
3 Governing Equations	15
3.1 Fluid Conservation Laws	15
3.1.1 Farve-Averaged Navier-Stokes Equations	17
3.1.2 Boussinesq Approximation	19
3.2 Spalart-Allmaras Turbulence Closure	20
4 Numerical Methods	26
4.1 Strand Grids Discretization Method	26
4.2 Discretization of FANS and SA Turbulence Model System	29
4.2.1 Inviscid Flux	30
4.2.2 Viscous Flux	30
4.2.3 Source Terms	32
4.3 Boundary Conditions and Boundary Discretization	33
4.4 Method of Manufactured Solutions	36

5	Results and Discussion	39
5.1	Verification	39
5.2	Validation	42
5.2.1	Flat Plate with Zero Pressure Gradient	42
5.2.2	Bump-in-channel	44
5.2.3	NACA 0012 Airfoil	49
5.2.4	NACA 4412 Trailing Edge Separation	54
6	Conclusions	61
6.1	Conclusions	61
6.2	Future Work	62
	References	64

List of Tables

Table		Page
5.1	Comparison of computed drag coefficients for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$	44
5.2	Comparison of computed drag coefficients for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$	49
5.3	Comparison of computed lift and drag coefficients for flow over a NACA 0012 airfoil at $M = 0.15$, $\alpha = 15^\circ$ $Re = 6 \times 10^6$. Note that the FUN3D and CFL3D results use a much finer grid.	50
5.4	Comparison of computed lift and drag coefficients for flow over a NACA 4412 airfoil. at $M = 0.09$, $\alpha = 13.87^\circ$ $Re = 1.52 \times 10^6$	58

List of Figures

Figure		Page
1.1	Strand grid elements and example strand-Cartesian grid system for the TRAM rotor.	2
2.1	Sample strand meshes around a square cylinder to illustrate how smoothing affects strands [1].	7
2.2	Example of internal corner with smoothing and clipping [1].	8
2.3	Observations from Reynolds' Dye and Pipe Experiment [2].	8
4.1	A sample 2D mesh with cells c1 and c2. k represents the face between the cells.	27
4.2	A sample 1D cell centered boundary with extrapolated point and ghost node.	34
5.1	MMS grid refinement study results.	39
5.2	Various meshing strategies used for manufactured solution grid refinement studies.	40
5.3	RMS density error for manufactured solution grid refinement studies using various meshing strategies on isotropic and stretched strand distributions.	41
5.4	Flat Plate Dimensions and Boundary Conditions [3]	43
5.5	Grid and turbulent viscosity contours for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$	44
5.6	Comparison of streamwise velocity and turbulent viscosity profiles for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$	45
5.7	Contours of $\rho\tilde{\nu}$	46
5.8	Bump-in-channel Dimensions and Boundary Conditions [3]	46
5.9	Grid and turbulent viscosity contours for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$	47
5.10	Comparison of streamwise velocity and turbulent viscosity profiles for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$	48

5.11 Bump-in-channel surface coefficient of pressure and friction	48
5.12 Grid system for flow over a NACA 0012 airfoil.	50
5.13 Various meshing strategies for the trailing edge of a NACA 0012 airfoil. . .	51
5.14 Pressure coefficient for flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$ at various angles of attack. Only multi-strand results with no wake Cartesian refinement are shown. Other meshes were indistinguishable.	52
5.15 C_l vs. α compared to experiment using various meshing strategies for flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$	53
5.16 C_l vs. C_d compared to experiment using various meshing strategies for flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$	54
5.17 Field plots of velocity magnitude for various α , showing the effect of Cartesian refinement in the wake region. All cases use smoothed strands.	55
5.18 Grid system for flow over a NACA 4412 airfoil.	56
5.19 Velocity magnitude contour plot of the NACA 4412 airfoil	57
5.20 Pressure coefficient for flow over a NACA 4412 airfoil at $M = 0.09$ and $Re = 1.52 \times 10^6$ at an angle of attack of 13.87°	58
5.22 Direction field plot of the trailing edge separation over the NACA 4412 airfoil	59
5.21 Velocity profiles along the NACA 4412 trailing edge	60

Chapter 1

Introduction

One of the remaining bottlenecks in computational fluid dynamics (CFD) is viscous mesh generation for complex geometry. With current practices, meshing experts can spend days or weeks generating high-quality viscous meshes around complex multi-body geometry, such as rotorcraft. With projected improvements in hardware, the percentage of time devoted to mesh generation using current practices will only increase relative to total simulation time. This lack of automation places a heavy burden on CFD practitioners and severely limits the practical use of CFD for design and analysis of complex systems. Furthermore, these complex systems require ever-increasing mesh sizes, for which scalability becomes a greater issue. Automating viscous mesh generation and maintaining computational efficiency, while preserving spatial and temporal accuracy are currently among the greatest research challenges in CFD today.

The strand-Cartesian approach has shown great potential to alleviate many of these difficulties [4–6]. Strand and Cartesian grids allow the possibility of fully automatic volume grid generation while enhancing scalability and the potential for high-order accuracy. Near solid bodies, the strand approach automatically creates a prismatic mesh along “strands” emanating from pointing vectors determined from a surface tessellation in order to resolve viscous boundary layers and other near-body effects, as shown in Figure 1.1(a). Away from solid bodies, adaptive Cartesian grids resolve vortical shedding and wake features with efficient high-order algorithms, shown in Figure 1.1(b). Due to the robust and automatic nature of the strand-Cartesian grid generation process, the technique is easily extensible to moving-body problems for which the grid can readily be regenerated at each time step. Strand and Cartesian grids communicate through implicit overset interpolation [7–9], which is greatly facilitated by the fact that the entire strand-Cartesian mesh system can be stored

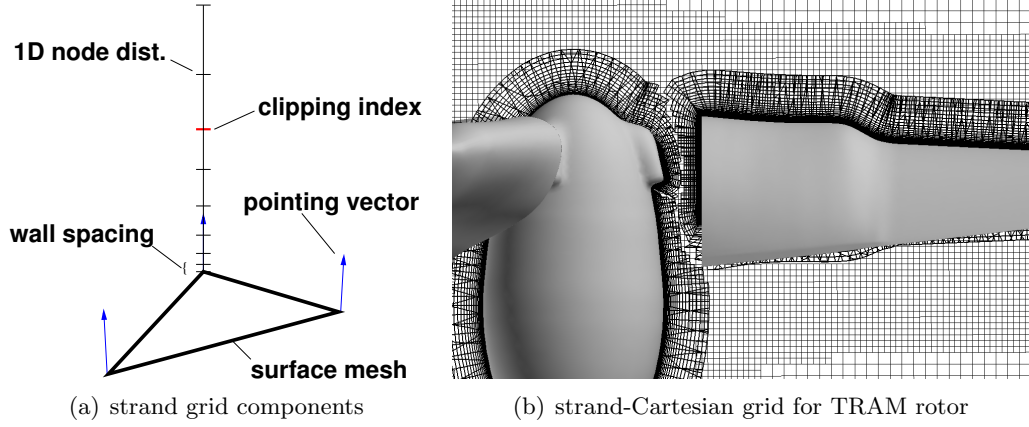


Fig. 1.1: Strand grid elements and example strand-Cartesian grid system for the TRAM rotor.

on each processor due to its compact grid representation. This allows for self-satisfying domain connectivity [4] and reduces the percentage of time needed for inter-grid communication [10].

The strand-Cartesian method has proven effective for both high and low Reynolds number complex flow and geometry [1,4] using various established solvers. A unique strand-Cartesian solver has been developed for the strand-Cartesian approach. This work develops this solver further to handle turbulent high Reynolds number flow for both simple and complex geometry. Modeling turbulence has been described by some as a “black hole” of research, because by its very nature, it is chaotic and unpredictable, thus making its prediction and simulation difficult. Nobel Laureate Richard Feynman famously described turbulence as “the most important unsolved problem of classical physics” [11]. While simulating turbulence is more often art than science, using current turbulence modeling methods, we may achieve good results that agree with experiment for many cases of engineering interest; provided the correct methods are chosen for the correct cases.

Currently, the most widely used methods for modeling turbulence consist of Direct Numerical Simulation (DNS), Large-Eddy Simulation (LES), and Reynolds-Averaged Navier-Stokes (RANS) or Farve-Averaged Navier-Stokes (FANS) turbulence models. The advance-

ment of more powerful hardware has made solution methods such as DNS and LES more desirable as they provide accurate yet computationally heavy solutions. Basic RANS and FANS turbulence models provide a way to predict flow regimes with some accuracy but with far lighter computational workload. This approach is consistent with the goals of extreme automation and computational efficiency targeted by the strand-Cartesian solver. A quick analysis of high Reynolds number flow over complex geometry could potentially be made in a fraction of the time current turbulence and meshing methods take. This type of quick analysis could be used as “first cut” over complex geometry before more accurate simulation methods such as DNS or LES are used to resolve the flow field.

This work aims to implement and validate the one-equation Spalart-Allmaras turbulence model for a strand-Cartesian solver. The current Navier-Stokes equations of motion will no longer be valid, and thus the Farve-Averaged Navier-Stokes equations of motion, coupled with the Spalart-Allmaras equation will now be implemented and solved in both the strand and Cartesian solvers. A variety of case studies are analyzed in order to verify and validate the SA model within the strand-Cartesian solver.

External sharp corners, such as those found on the trailing edge of an airfoil, present a challenge in meshing that is unique to strand-Cartesian methods. As strands emanate normal to the surface from a single node, sharp corners will obviously lack detailed resolution, which is critical for turbulent flow in order to capture flow separation and resolve the wake. Various methods such as strand smoothing, multi-strands, and overset Cartesian wake capturing meshes are investigated in this thesis.

1.1 Objectives

The objectives of this work are to:

- Investigate the new aspects of the Spalart-Allmaras turbulence model, and implement it with FANS as a coupled system of equations into the strand-Cartesian solver.
- Verify the mathematical accuracy of the Spalart-Allmaras model in the strand grid system. The Method of Manufactured Solutions shall be used to facilitate this.

- Validate this model using the validation cases provided by the NASA-Langley turbulence resource. The validation cases used are the; zero pressure gradient flat plate, bump-in-channel, NACA 0012 airfoil, and the NACA 4412 trailing edge separation.
- To examine different external sharp corner meshing strategies.

1.2 Outline of Thesis

This thesis will comprise of several chapters. Chapter 2 contains the Literary Review and will provide any background information that the reader may need for understanding this thesis. Chapter 3 provides the reader with the general theory and equations used for this work. Chapter 4 presents the numerical methods used in this work. Any key information to the reader about discretization and implementation methods are given here. Following this, the results of the verification and validation cases will be presented and discussed in Chapter 5. Chapter 6 will conclude the thesis and make recommendations for future work.

Chapter 2

Literary Review

This chapter provides the necessary background information, basic knowledge, and theory for the work performed in this thesis. A brief review of previous work done on strand grids, turbulence, turbulence modeling, and verification and validation is given below.

2.1 Strand Grids

Cartesian Adaptive Mesh Refinement (AMR) methods of grid generation can be made fully automatic and completely robust for arbitrary geometric configurations, however, Cartesian AMR methods cannot be used exclusively to satisfy HPC requirements for automation due to the length scales associated with viscous flow. The reason being, that Cartesian methods require isotropic refinement to accommodate arbitrarily complex boundary surfaces. Fluid/aerodynamic interactions with solid surfaces result in viscous boundary layers that require spacing too fine to be accommodated isotropically [4]. It is in these viscous boundary layers that strand grids are employed to provide the necessary grid resolution, resulting in the strand-Cartesian approach.

The strand-Cartesian grid approach provides many advantages for complex moving-body flow simulations, including fully-automatic volume grid generation, highly scalable domain connectivity, and high-order accuracy. Strand grids are created from discrete surface geometry comprised on n-sided polygons. Tessellated geometries can be formed from any n-sided polygon, including triangles and quadrilaterals in 3D. A strand is created for each vertex of the discretized surface geometry. A unit normal vector is approximated at each surface vertex by averaging the normals of neighboring polygons. Each strand is placed so its origin, or root, coincides with the vertex and initially points in the direction of the corresponding surface normal. Every strand in the mesh shares a length and user-defined

1D nodal distribution along the strand. Each strand has a clipping index to “shorten” the strand to prevent intersection with other bodies. This is to say that nodes that lay beyond the clipping index are not valid physical nodes [1,4]. Fig. 1.1(a) demonstrates this procedure on a simplified surface triangle.

Sharp corners, both internal and external, present challenges for strand grids. At internal corners, strands tend to intersect and penetrate the surface geometry. Strands typically fail to have sufficient resolution at external corners. Strand smoothing can help to mitigate the challenges presented by corners. Strand smoothing refers to a simple method of adjusting the surface normals to properly cover all regions of the domain. Fig. 2.1(a) shows a completely un-smoothed strand mesh around a square cylinder where Fig. 2.1(b) illustrates this same mesh after being smoothed.

An alternate approach to handling sharp corners has been investigated by Work [12] et. al. Multi-strands work by placing multiple stands on a single node at the point of the corner, and fanning strands evenly around this angle. Fanning strands in an even distribution around a sharp corner, while simple in 2D, provides a challenge in 3D. In 3D, the number of strands required from a single node, and the small angles that would separate the strands in order to provide the necessary resolution at the corner, provides numerous numerical and meshing issues such as; small unusual shaped high aspect ratio cells with many faces. The NACA 0012 case in Chapter 5 presents some of these results for a 2D airfoil. Strand smoothing with Cartesian refinement in the corners was shown to be the most effective method for resolving external corners.

Strand smoothing is typically insufficient for sharp internal corners. Even with smoothing, strands often intersect creating a negative volume element. Strand clipping is used to ensure that strand cells do not intersect or penetrate the surface geometry. An example of a smooth and clipped internal corner is in Fig. 2.2.

Strand grid technology has an advantage over other mesh technologies as it collapses memory requirements for near-body domain partitions from volume data to surface data only. Non-surface grid related data that may be required during a simulation is trivially

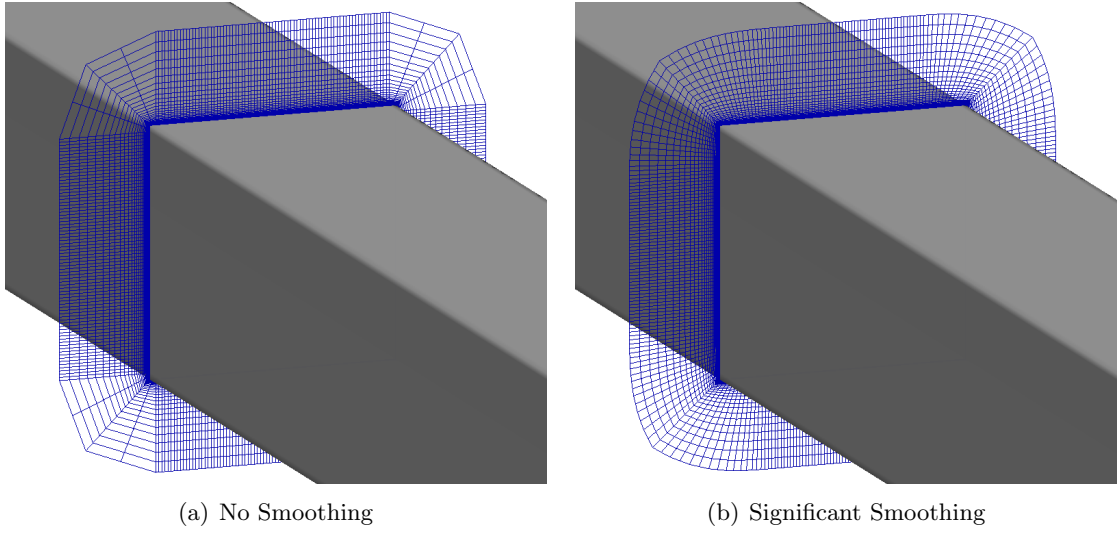


Fig. 2.1: Sample strand meshes around a square cylinder to illustrate how smoothing affects strands [1].

computed. It is easy to undervalue the advantage of minimal memory requirements for grid related data. After all, memory is relatively cheap. The advantage is realized in context of a design environment with HPC systems. Maximum solution throughput is obtained by allocating the minimum number of processors that can accommodate a given problem in core memory. Multiple cases can then be run simultaneously, realizing perfect parallelism [4].

2.2 Turbulence

In 1883 Reynolds [2] published the results from a series of experiments in order to characterize the different flow regimes. The experiment consisted of dye in a pipe that contained different speeds of water flowing through it. In slow water the dye stayed streamline. As the speed of the water increased, the dye became progressively unstable. Figure 2.3 shows a diagram of these findings.

Figure 2.3(a) shows the streamline laminar flow, while Figure 2.3(b) shows the eventual turbulent flow that occurs once a critical velocity has been reached. Reynolds also observed that if the critical velocity was slowly reached, there was an emergence of “eddies,” a key feature to turbulent flow, shown in Figure 2.3(c).

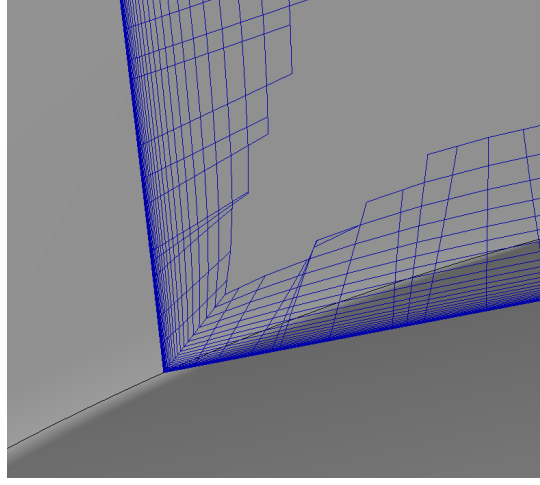


Fig. 2.2: Example of internal corner with smoothing and clipping [1].

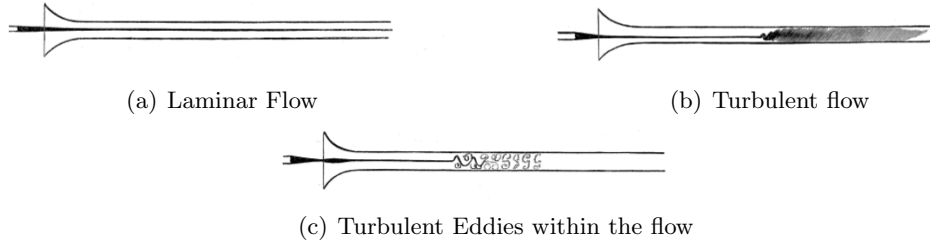


Fig. 2.3: Observations from Reynolds' Dye and Pipe Experiment [2].

The end result of these experiments was the introduction of a dimensionless quantity; the Reynolds number (Re). It is defined as the ratio of inertia forces to viscous forces. It is used to characterize different flow regimes within a similar fluid, such as laminar or turbulent flow: laminar flow occurs at low Reynolds numbers, where viscous forces are dominant, and is characterized by smooth, constant fluid motion; turbulent flow occurs at high Reynolds numbers and is dominated by inertial forces, which tend to produce chaotic eddies, vortices and other flow instabilities.

There is no general theorem that relates Reynolds number and turbulence. For each type of flow, a different Reynolds number is required for turbulence. For example, Poiseuille flow with a Reynolds number value over 2300 begins to become turbulent. Turbulence flow is

generally interspersed with laminar flow until a larger Reynolds number of about 4000 [13].

In spite of how common turbulence is, it is not easy to define precisely, however, turbulence has certain characteristics, that Kundu [14] describes as:

- *Randomness*: Turbulent flow is irregular, unpredictable and chaotic.
- *Nonlinearity*: Turbulent flows are highly nonlinear. The nonlinearity serves two purposes. First it causes the relevant nonlinearity parameter to exceed critical value. In unstable flows small perturbations grow spontaneously and frequently equilibrate as finite amplitude disturbance. On further exceeding the stability criteria, the new state can become unstable to more complicated disturbances, and the flow eventually reaches a chaotic state. Second, the nonlinearity of a turbulent flow results in vortex shedding, a key process by which three-dimensional turbulent flows maintain their vorticity.
- *Diffusivity*: Turbulent flows have a rapid rate of diffusion of heat and momentum due to the macroscopic mixing of fluid particles.
- *Vorticity*: Turbulent flows are characterized by high levels of fluctuating vorticity. The identifiable structures in a turbulent flow are vaguely called *eddies*. Flow visualization of turbulent flows shows various structures - coalescing, dividing, stretching, and above all spinning. A characteristic feature of turbulence is the existence of an enormous range of eddy sizes. The large eddies have a size of order roughly equivalent to the width of the region of the turbulent flow; in a boundary layer this is the thickness of the layer. The large eddies contain most of the energy. The energy is handed down from large to small eddies by nonlinear interactions, until it is dissipated by viscous diffusion in the smallest eddies, whose size is of the order of millimeters.
- *Dissipation*: The vortex stretching mechanism transfers energy and vorticity to increasingly smaller scales, until the gradients become so large that they are smeared out (i.e. dissipated) by viscosity. Turbulent flows therefore require a continuous supply of energy to make up for the viscous losses.

In 1941 Kolmogorov [15] postulated that for very high Reynolds numbers, the small scale turbulent motions are statistically isotropic, thus Kolmogorov length scales were introduced. Kolmogorov length scales are the smallest scales in the spectrum that form the viscous sub-layer range. In this range, the energy input from nonlinear interactions and the energy drain from viscous dissipation are in exact balance. The small scales are in high frequency, which is why turbulence is locally isotropic and homogeneous.

Laminar flow may easily be modeled by simply solving the Navier-Stokes equations, turbulent flow provides a challenge however. Using ordinary CFD techniques (not DNS or LES), it is possible to find some particular solutions of the Navier-Stokes equations governing fluid motion, however, all such solutions are unstable to finite perturbations at large Reynolds numbers. Thus different CFD methods must be used (DNS or LES), or some type of turbulence modeling must be utilized.

2.3 Turbulence Modeling and One-Equation Turbulence Models

2.3.1 Turbulence Modeling

Direct numerical simulation is potentially the most accurate method of modeling turbulence. DNS is a CFD simulation that solves the Navier-Stokes equations. This requires that the entire range of temporal and spatial scales of the turbulence to be solved. In the computational mesh, all spatial scales must be resolved [16]; from the smallest dissipative scales (Kolmogorov scales), to the integral scale L , which is associated with the motions containing most of the kinetic energy. Due to range of scales that need be resolved, DNS requires powerful hardware to simulate.

Because of the computational cost, DNS is used as a tool in the fundamental research of turbulence. DNS is used to better understand the physics of turbulence through the use of “numerical experiments.” Generally the information extracted is difficult or impossible to obtain in a laboratory. DNS may also be used in the development of turbulence models for practical applications, such as those used in the subgrid in LES models, and methods that solve the RANS equations, such as the SA model.

A practical and accurate tool for simulating turbulent flows is large-eddy simulation (LES). In turbulent flow, the large eddies are dependent on geometry, while the smaller scales are more universal [15]. The velocity field in turbulent flow may be separated into resolved and subgrid components. The resolved portion of the field represents the “large eddies,” while the subgrid part of the field represents the “small eddies,” whose effect on the resolved field is included through the subgrid scale model. This feature of turbulent flow allows for large eddies to be solved explicitly in calculation. The small eddies are implicitly accounted for by using a subgrid-scale model. Subgrid-scale (SGS) modeling [17] is used to represent the effects of unresolved small-scale fluid motions (small eddies, swirls, vortices) in the equations governing the large-scale motions that are resolved in computer models.

Both DNS and LES provide the most accurate results when simulating turbulence, but at the expense of an extremely high computation workload. A computationally lighter alternative to DNS and LES is to use a RANS turbulence model. A turbulence model aims to model all the unsteady turbulence eddies. No attempt is made to resolve the unsteady features of any of the turbulent eddies, not even the largest ones. Instead, mathematical models are employed to take into account the enhanced mixing and diffusion caused by turbulent eddies. When using a turbulence model, the Navier-Stokes equations are no longer employed, instead the Favre-Averaged Navier-Stokes equations are used (Chapter 3).

2.3.2 One-Equation Turbulence Models

One-equation turbulence models solve one turbulent transport equation in conjunction with the RANS or FANS equations. One turbulent transport term, usually the turbulent kinetic energy, is solved by the turbulent equation. The original one-equation model is Prandtl’s one-equation model [18]. The Spalart-Allmaras model, which is the focal model in this work, is based on Prandtl’s original one-equation model. Other one-equation models exist, such as the Rahman-Siikonen-Agarwal Model and the Baldwin-Barth model, however these do not share the popularity of the SA model, especially for external aerodynamics.

The standard Spalart-Allmaras model [19] consists of a single transport equation which

calculates the kinematic eddy viscosity ($\tilde{\nu}$). This model has been shown to give good results for adverse pressure gradients and boundary layers. Thus this model has become popular for aerodynamic and turbo-machinery purposes [20]. The original Spalart-Allmaras equation has the basic form of:

$$\rho \frac{\partial(\tilde{\nu})}{\partial t} + C = F + P - D + T \quad (2.1)$$

Here we define C as the convection of $\tilde{\nu}$, F as the diffusion of $\tilde{\nu}$, P as the rate of production of $\tilde{\nu}$, D as the rate of dissipation of $\tilde{\nu}$, and T as another term for the transport of $\tilde{\nu}$ by turbulent diffusion. For convenience, we may let $P - D + T = S$, and call these the source terms. Chapter 3 provides a more explicit explanation of the model and its modifications.

The advantage to a one-equation model is simply that there is only one additional equation to solve. Multi-equation models require two or more additional equations to be solved. While multi-equation models may or may not produce better results, there is added complexity when implementing the model into the CFD solver, such as linearizing the source terms. Adding multiple-equation turbulence models to the equations of motion adds extra time required for the solution to converge. Chapter 4 describes the methods required for discretizing and implementing the turbulence model.

2.4 Verification and Validation

2.4.1 Verification

Whenever new numerical algorithms are designed and implemented in a CFD code, it is necessary to perform detailed verification to ensure accuracy and to test the code's integrity. Verification in this sense is defined as:

The process of determining that a model's implementation accurately represents the developer's conceptual description of the model and the solution to the model. [21]

Roache [22], Roy [23], and Veluri [24] showed how the methods of manufactured solutions (MMS) can be used to verify the desired order of accuracy of the scheme with the actual order of accuracy the code produces. They demonstrated that verification using MMS is capable of identifying malformed algorithms and general coding “bugs.”

Many researchers have extended MMS to verify complex problems. Diskin et al. [25–27] have employed MMS extensively to compare and test different second order schemes as well as study the effect irregular grids have on accuracy. Many others have developed MMS methodologies for the interior solution only [28–33], while Folkner and Katz [34] investigated this methodology on boundary conditions.

Roache [22] describes a general MMS provides a general procedure for working with such analytical solutions. The procedure is very simple. A continuum solution is constructed, which in general will not satisfy the governing equations. An appropriate source term can be determined to cancel any imbalance in the PDEs caused by the choice of the continuum solution. The solution also defines the boundary conditions in all forms, be they Dirichlet, Neumann or Robin. The chosen solution need not have a physical meaning since verification (of codes or of calculations) is a purely mathematical exercise. But choosing a physically realistic manufactured problem which has a closed form solution offers a useful advantage: It exercises each term involved in the PDEs in a manner similar to that of a real problem so that similar difficulties in the solution and error estimation processes will arise.

Once the manufactured solution has been constructed and the source terms determined for the set of equations to be verified, code verification can take place on any grid in the domain covered by the MMS. By verifying the code on increasing grid resolutions, we may show that as grid resolution is increased, the solution becomes more accurate.

2.4.2 Validation

Once a CFD code has been verified, it is important to validate the code’s application to certain problems. A major goal of this thesis is to validate the Spalart-Allmaras model for strand grids. Validation of this model within the strand grids framework is important

as many new aspects of simulating turbulence are investigated, primarily different strand grid meshing strategies to best resolve the flow field around the trailing edge of an airfoil and to capture the wake.

Validation may be described as “solving the right equations.” It is not possible to validate the entire CFD code. One can only validate the code for a specific range of applications for which there is experimental data. Thus one validates a model or simulation. Applying the code to flows beyond the region of validity is thus termed prediction. A more precise definition of validation in this scenario is given as:

The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. [21]

Validation examines if the conceptual models, computational models as implemented into the CFD code, and computational simulation agree with real world observations. The strategy is to identify and quantify error and uncertainty through comparison of simulation results with experimental data. The experiment data sets themselves will contain bias errors and random errors which must be properly quantified and documented as part of the data set. The accuracy required in the validation activities is dependent on the application, and so, the validation should be flexible to allow various levels of accuracy.

Chapter 3

Governing Equations

This chapter describes the necessary equations needed to model turbulent flow in the strand grids framework. The general equations of motion for a fluid will be described first. How these equations may be altered to accommodate turbulence modeling will follow. The Spalart-Allmaras model will then be described, with details of how these were altered for CFD purposes.

3.1 Fluid Conservation Laws

To accurately model laminar two dimensional flow in CFD, four equations must be solved; mass, x-momentum, y-momentum and energy. These equations in index notation are shown below in Strong Conservation Law (SCL) form in equations 3.1 to 3.3.

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = 0 \quad (3.1)$$

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_j u_i + p \delta_{ij})}{\partial x_j} + \frac{\partial(\sigma_{ij})}{\partial x_j} + \rho g_i = 0 \quad (3.2)$$

$$\frac{\partial(\rho e)}{\partial t} + \frac{\partial(\rho h u_j)}{\partial x_j} - \frac{\partial(\sigma_{ij})}{\partial x_j} u_i - \frac{\partial q_j}{\partial x_j} - \rho g_j u_j = 0 \quad (3.3)$$

Here we shall define p as pressure, g_i as the i th component of the body force, e as the total energy (internal and kinetic) per unit mass and h as enthalpy; which may also be defined as:

$$h = e + \frac{p}{\rho} \quad (3.4)$$

The variable q_j is the j -th component of the heat flux vector. We can relate the heat flux vector to the temperature gradient through Fourier's law of heat conduction:

$$q_j = -\kappa_q \frac{\partial T}{\partial x_j} \quad (3.5)$$

$$\kappa_q = \frac{C_p \mu}{Pr} = \frac{\gamma R \mu}{(\gamma-1)Pr}$$

R is the gas constant for the fluid in use, γ is the ratio of specific heats, Pr is the Prandtl number and κ_q is the coefficient of thermal conductivity.

Up to this point, the only assumptions we have made are that the fluid is a continuum, in which we neglect the true molecular nature of matter. We will make two other assumptions about the equations of motion. First, we will neglect body forces, such as gravity. This is a good approximation for isothermal fluids with low density, such as air. Second, we will assume that the fluid is “Newtonian,” which means that the rate of strain of the fluid is linearly proportional to the stress. The classic Newtonian stress tensor (σ_{ij}) has the form [35]:

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \right) \quad (3.6)$$

From a CFD perspective, these equations may be written in a more compact form as:

$$\frac{\partial Q}{\partial \tau} + \frac{\partial F_j}{\partial x_j} - \frac{\partial F_j^v}{\partial x_j} = 0 \quad (3.7)$$

The vectors Q , F_j and F_j^v are defined as:

$$Q = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \end{pmatrix}, \quad F_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + \delta_{ij} p \\ \rho h u_j \end{pmatrix}, \quad F_j^v = \begin{pmatrix} 0 \\ \sigma_{ij} \\ \sigma_{ij} u_i - q_j \end{pmatrix} \quad (3.8)$$

Here, Q is the vector of conserved variables. These are the dependent variables that are solved. F_j is known as the inviscid flux vector, and F_j^v is known as the viscous flux vector. If we set $F_j^v = 0$, we obtain the Euler equations, which is a model for “inviscid flow.” The Euler equations provide a useful “first cut” approximation.

Using ordinary CFD techniques, i.e. not DNS or LES, the solution to these equations provide an accurate representation of the physical problem in laminar flow; however, if the Reynolds number is high enough such that the flow becomes turbulent, then the laminar solution will no longer be accurate or appropriate. Thus to accurately model turbulent flow, these equations of motions must be modified to account for turbulence.

3.1.1 Farve-Averaged Navier-Stokes Equations

In this section we aim to look at both the Reynold-Averaged Navier-Stokes (RANS) equations and the Farve-Averaged Navier-Stokes equations (FANS). Generally the RANS equations are appropriate for most applications, and the FANS equations used only when encountering highly compressible or hypersonic flow. As the strand-Cartesian approach is designed for aerodynamic flows, compressible flows will be encountered, thus we use FANS equations. RANS turbulence models are equally applicable to FANS equations.

Before we can modify the equations of motion, some notation must first be introduced. We introduce classic time averaging (Reynolds averaging) and density weighted time averaging (Farve averaging). Both systems are broken down into mean and fluctuations. The mean is donated by either an overbar or tilde, and fluctuations by either a prime or double prime. Reynolds averaging for some variable Φ is defined as:

$$\begin{aligned}\overline{\Phi} &= \frac{1}{T} \int_T \Phi(t) dt \\ \Phi &= \overline{\Phi} + \Phi'\end{aligned}\tag{3.9}$$

Farve averaging for some variable Φ is defined as:

$$\begin{aligned}\tilde{\Phi} &= \frac{\rho\Phi}{\bar{\rho}} \\ \Phi &= \tilde{\Phi} + \Phi''\end{aligned}\tag{3.10}$$

It is important to note that the mean of the Reynolds fluctuations is zero, however, this is does not hold true for Farve fluctuations.

$$\begin{aligned}\overline{\Phi'} &= 0 \\ \overline{\Phi''} &\neq 0\end{aligned}\tag{3.11}$$

Applying both Reynolds and Farve averaging to the equations of motion, there now exists a new set of equations which account for turbulent fluctuations. Oliver [36] describes this in great detail. Taking the final result we obtain an open set of equations that is commonly known as the Farve-Averaged Navier-Stokes (FANS) equations. These equations in SCL form with our prior assumptions have the form of:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_j)}{\partial x_j} = 0\tag{3.12}$$

$$\frac{\partial(\bar{\rho}\tilde{u}_i)}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_j\tilde{u}_i - \bar{p}\delta_{ij})}{\partial x_j} + \frac{\partial}{\partial x_j}(\bar{\sigma}_{ij} - \overline{\rho u_i'' u_j''}) = 0\tag{3.13}$$

$$\frac{\partial(\bar{\rho}\tilde{E})}{\partial t} + \frac{\partial(\bar{\rho}\tilde{H}\tilde{u}_j)}{\partial x_j} - \frac{\partial}{\partial x_j}[(\bar{\sigma}_{ij}\tilde{u}_i - \overline{\rho u_i'' u_j''})] - \frac{\partial}{\partial x_j}(\bar{q}_j + \overline{\rho u_j'' h''}) + \frac{\partial}{\partial x_j}(\bar{\sigma}_{ij} u_i'' - \frac{1}{2} \overline{\rho u_j'' u_i'' u_i''}) = 0\tag{3.14}$$

Immediately we can identify the effects of turbulence in these equations: the Reynolds stress tensor $-\overline{\rho u_i'' u_j''}$ and the Reynolds heat flux $\overline{\rho u_j'' h''}$. Additionally, we see two terms that refer to turbulent transport and work: transport of turbulent kinetic energy by turbulent velocity fluctuations $-\frac{1}{2} \overline{\rho u_j'' u_i'' u_i''}$ and work done by the viscous stress due to turbulent velocity fluctuations $\bar{\sigma}_{ij} u_i''$.

We define some turbulent terms. Total turbulent energy is defined as E (total energy and turbulent kinetic energy), H is the total turbulent enthalpy, k is the turbulent kinetic energy, μ_t is the turbulent eddy viscosity and q_{Tj} is the turbulent heat flux. We also define the transport and work of turbulence [36]. These relationships are shown below:

$$E = e + k\tag{3.15}$$

$$H = h + k\tag{3.16}$$

$$\overline{\rho u_j'' h''} = q_{Tj} = -\frac{C_p \mu_t}{Pr_T} \frac{\partial T}{\partial x_j} \quad (3.17)$$

$$\frac{\partial}{\partial x_j} (\overline{\sigma_{ij} u_i''} - \frac{1}{2} \overline{\rho u_j'' u_i'' u_i''}) = (\mu + \frac{1}{2} \mu_t) \frac{\partial k}{\partial x_j} \quad (3.18)$$

The turbulent Prandtl number is Pr_T , and often has a value of 0.9 for air. For comparison, the Prandtl number for air often has a value of 0.75. This form of the Navier-Stokes can be manipulated into a vector form, resulting in [37]:

$$Q = \begin{pmatrix} \rho \\ \rho u_i \\ \rho E \end{pmatrix}, \quad F_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + \delta_{ij} p \\ \rho H u_j \end{pmatrix}, \quad F_j^v = \begin{pmatrix} 0 \\ \sigma_{ij} - \overline{\rho u_i'' u_j''} \\ u_i (\sigma_{ij} - \overline{\rho u_i'' u_j''}) + (\mu + \frac{1}{2} \mu_t) \frac{\partial k}{\partial x_j} - (q_j + q_{Tj}) \end{pmatrix} \quad (3.19)$$

Turbulence models are based on solving variables so that the Reynolds stress tensor may be given a value. A note should be made, that from this point on, the overbar and tilde on singular mean variable terms shall be implied. Fluctuating terms shall still retain their prime or primes. Many turbulence methods utilize the Boussinesq Approximation in order to solve for this term.

3.1.2 Boussinesq Approximation

The Boussinesq Approximation was proposed by Boussinesq in 1877 [38] and hypothesized that Reynolds stresses might be proportional to the mean rate of deformation. This hypothesis introduces the concept of eddy viscosity (μ_t). Eddy viscosity describes the dissipation of momentum caused by turbulent eddies, and is analogous to the dissipation of momentum at the molecular level caused by a fluid's viscosity.

$$\tau_{ij} \equiv -\overline{\rho u_i'' u_j''} = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij} \quad (3.20)$$

The turbulent kinetic energy is symbolized by k , which is a base concept for many

turbulence models, most notably two-equation models. The disadvantage of the Boussinesq hypothesis as presented is that it assumes μ_t is an isotropic scalar quantity, which is not strictly true. The Reynolds stress tensor shall be notated by τ_{ij} from this point on to keep notation compact. We reach an equation with the final form of:

$$Q = \begin{pmatrix} \rho \\ \rho u_i \\ \rho E \end{pmatrix}, F_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + \delta_{ij} p \\ \rho H u_j \end{pmatrix}, F_j^v = \begin{pmatrix} 0 \\ \sigma_{ij} + \tau_{ij} \\ u_i(\sigma_{ij} + \tau_{ij}) + (\mu + \frac{1}{2}\mu_t)\frac{\partial k}{\partial x_j} - (q_j + q_{Tj}) \end{pmatrix} \quad (3.21)$$

In order to reduce notation, from this point we shall let $\sigma_{ij} = \sigma_{ij} + \tau_{ij}$ and $q_j = q_j + q_{Tj}$.

3.2 Spalart-Allmaras Turbulence Closure

In Chapter 2, the general form of the Spalart-Allmaras equation was shown and discussed. In this section, more precise definitions are given for the SA model. The original Spalart-Allmaras equation has the form of [19]:

$$\rho \frac{D\tilde{\nu}}{Dt} = \rho C_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} - [\rho C_{\omega 1} f_{\omega} - \rho \frac{C_{b1}}{\kappa^2} f_{t2}](\frac{\tilde{\nu}}{d})^2 + \frac{1}{\sigma} [\frac{\partial}{\partial x_j}((\mu + \rho\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}) + C_{b2}\rho \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \tilde{\nu}}{\partial x_k}] \quad (3.22)$$

Since the arrival of the original Spalart-Allmaras model, a number of modifications to the model have been made, creating new and alternate SA models. Some of these are simple changes, such as setting trip terms to zero. Other changes are more drastic such as those proposed by Catris [39]. The emergence of all these modified SA models has created some confusion when selecting which form of the SA model to use.

One of the reasons that SA models were altered was because situations arise on coarse grids and transient states where the turbulent solution may become negative. This is often encountered at the edge of the boundary layers and wakes, where the turbulence solution is characterized by ramp solutions that transition to constant outer/freestream regions over

a short region. The rapid transition from large inner to relatively small outer levels can result in undershoots for discrete solutions. These undershoots may cross zero, resulting in a negative solution values. Previous methods of eliminating negative solution values has been to clip updates.

Recently Allmaras has added some much needed clarity to the selection of SA models, by proposing a new “negative model” [40], which is to be used in conjunction with the original SA model (or another SA model if the user so desires). The negative SA model corrects these discrete undershoots that cross zero, and gives the SA model robustness. In this work, the original SA model (with trip terms) shall be used when $\tilde{\nu} > 0$. When $\tilde{\nu} < 0$ the negative model shall be utilized. Allmaras’s most recent work makes a minor modification to the original proposed SA model. This modification accommodates compressible flow by not holding the density constant in the turbulent diffusion coefficient η (see below). By not holding the density constant the SA model becomes more accurate and robust in compressible flow. This extra term is absorbed into T (see below).

To make any SA model usable in CFD, some minor alterations must be made. The first modification we must make is to use the chain rule to make the LHS of the SA equation comply with SCL form. This modification yields:

$$\rho \frac{D\tilde{\nu}}{Dt} = \frac{\partial(\rho\tilde{\nu})}{\partial t} + \frac{\partial(\rho u_j \tilde{\nu})}{\partial x_j} \quad (3.23)$$

This lone modifications yields the combined standard-negative model below:

$$\frac{\partial(\rho\tilde{\nu})}{\partial t} + \frac{\partial(\rho u_j \tilde{\nu})}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\frac{\eta}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_j} \right) + P - D + T \quad (3.24)$$

The source terms P , D , and T are defined explicitly below. P , the rate of production of $\tilde{\nu}$, is simply known as the production term.

$$P = \begin{cases} \rho C_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu}, & \tilde{\nu} \geq 0 \\ \rho C_{b1}(1 - C_{t3})\Omega\tilde{\nu}, & \tilde{\nu} < 0 \end{cases} \quad (3.25)$$

The term D is defined as the rate of dissipation of $\tilde{\nu}$, also know as the destruction term.

$$D = \begin{cases} \rho[C_{\omega 1} f_{\omega} - \frac{C_{b1}}{\kappa^2} f_{t2}] (\frac{\tilde{\nu}}{d})^2, & \tilde{\nu} \geq 0 \\ -\rho C_{\omega 1} (\frac{\tilde{\nu}}{d})^2, & \tilde{\nu} < 0 \end{cases} \quad (3.26)$$

It is important to note that when $\tilde{\nu} < 0$, the destruction term D become positive. Below T is defined. Commonly known as the transport term, it is a term for the transport of $\tilde{\nu}$ by turbulent diffusion.

$$T = \frac{\rho C_{b2}}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \tilde{\nu}}{\partial x_k} - \frac{(\nu + \tilde{\nu} f n)}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \rho}{\partial x_k} \quad (3.27)$$

The turbulence diffusion coefficient η , is defined as:

$$\eta = \mu + \rho \tilde{\nu} f n \quad (3.28)$$

$$f n = \begin{cases} 1, & \tilde{\nu} \geq 0 \\ \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}, & \tilde{\nu} < 0 \end{cases} \quad (3.29)$$

Dynamic eddy viscosity may be related to $\tilde{\nu}$ via Eq 3.30. When $\tilde{\nu}$ is negative, the eddy viscosity is set to zero, and $\tilde{\nu}$ itself becomes a passive scalar. Using this relationship, the Reynolds stress tensor may be described as below:

$$\mu_t = \begin{cases} \rho \tilde{\nu} f_{v1}, & \tilde{\nu} \geq 0 \\ 0, & \tilde{\nu} < 0 \end{cases} \quad (3.30)$$

$$\tau_{ij} = \rho \tilde{\nu} f_{v1} (\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij}) - \frac{2}{3} \rho k \delta_{ij} \quad (3.31)$$

The turbulent kinetic energy variable is not calculated in the Spalart-Allmaras model, and thus the last term is often ignored when estimating Reynolds stresses.

Wall Damping Functions and Closure Coefficients

Simple eddy-viscosity models (EVM) such as SA rely on damping functions to mimic the attenuation of the turbulence near the wall. These damping functions depend on the local turbulent Re number or wall distance. The SA model contains some wall damping functions, such as the one mentioned in Eq 3.30. The wall damping function f_{v1} has two purposes. At the wall, the damping function f_{v1} becomes zero. At high Reynolds numbers the function tends to unity, and thus kinematic eddy viscosity $\tilde{\nu}$ simply becomes the kinematic eddy viscosity ν_t .

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}} \quad (3.32)$$

$$\chi = \frac{\tilde{\nu}}{\nu}$$

The wall damping function f_{v2} serves as similar purpose to f_{v1} . At the wall the damping function becomes unity, and at high Re the function becomes zero.

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \quad (3.33)$$

Another wall function used is the non-dimensional function f_w . In the logarithmic layer of a zero pressure gradient boundary layer $f_w \simeq 1$.

$$f_w = g \left(\frac{1 + C_{\omega 3}^6}{g^6 + C_{\omega 3}^6} \right)^{\frac{1}{6}} \quad (3.34)$$

$$g = r + C_{\omega 2}(r^6 + r)$$

$$r = \frac{\tilde{\nu}}{\Omega \kappa^2 d^2}$$

d is the distance from the current point to the nearest wall and Ω is the magnitude of the vorticity, commonly defined by:

$$\Omega = \sqrt{2\Omega_{ij}\Omega_{ij}} \quad (3.35)$$

$$\Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right)$$

$$\hat{S} = \begin{cases} \Omega + \bar{S}, & \bar{S} \geq -c_2\Omega \\ \Omega + \frac{\Omega(c_2^2 + c_3\bar{S})}{(c_3 - 2c_2)\Omega - \bar{S}}, & \bar{S} < -c_2\Omega \end{cases}, \quad (3.36)$$

$$\bar{S} = \frac{\tilde{\nu} f_{v2}}{\kappa^2 d^2}, \quad (3.37)$$

It is important to note that Ω must not be allowed to reach zero in order to avoid numerical problems, thus this particular form of \hat{S} from Oliver [41] is used to avoid this. The closure coefficients suggested by Allmaras [19, 40] are as follows:

$$cb_1 = 0.1355, \quad cb_2 = 0.622, \quad \sigma = \frac{2}{3}, \quad \kappa = 0.41$$

$$c_{\omega 2} = 0.2, \quad c_{\omega 3} = 2, \quad c_{v1} = 7.1 \quad c_{t1} = 1, \quad c_{t2} = 2, \quad c_{t3} = 1.2, \quad c_{t4} = 0.5 \quad (3.38)$$

$$c_{\omega 1} = \frac{cb_1}{\kappa^2} + \frac{1+cb_2}{\sigma}$$

Final SA Form Implemented in Strand Grids

Keeping with a system of equation notation, the Spalart-Allmaras model and the FANS equations of motion may be written as:

$$\frac{\partial Q}{\partial \tau} + \frac{\partial F_j}{\partial x_j} - \frac{\partial F_j^v}{\partial x_j} = S \quad (3.39)$$

The turbulent kinetic energy k has been ignored in the FANS energy equation (Eq. 3.14), since SA is a one-equation model, k is not calculable. The vectors Q , F_j , F_j^v , and S for the standard-negative SA model are now defined as:

$$Q = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \\ \rho \tilde{\nu} \end{pmatrix}, \quad F_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + \delta_{ij} p \\ \rho u_j h \\ \rho u_j \tilde{\nu} \end{pmatrix}, \quad F_j^v = \begin{pmatrix} 0 \\ \sigma_{ij} \\ \sigma_{ij} u_i - q_j \\ \frac{\eta}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_j} \end{pmatrix}, \quad S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ P - D + T \end{pmatrix} \quad (3.40)$$

Chapter 4

Numerical Methods

This chapter will demonstrate the methods in which the governing equations provided in Chapter 3 were discretized in order to be applied to strand grids. Note that in the strand-Cartesian solver, the equations presented in 3.39 are solved as coupled equations. While this makes the implementation and solving of the equations more expensive, it accounts for all cross-coupled terms.

4.1 Strand Grids Discretization Method

In order to discretize Eq. 3.39 using the finite volume method, it is necessary to integrate over the volume of the problem.

$$\int_V \left(\frac{\partial Q}{\partial t} + \frac{\partial F_j}{\partial x_j} - \frac{\partial F_j^v}{\partial x_j} = S \right) dV \quad (4.1)$$

This integral can be simplified using the divergence theorem (often referred to as Gauss's theorem) [42]. This theorem states that the flux through the boundary is equal to the sources or sinks of flux in the closed domain. This yields:

$$\int_V \frac{\partial Q}{\partial t} dV + \int_A F_j n_j dA - \int_A F_j^v n_j dA = \int_V S dV \quad (4.2)$$

It is important to maintain that integrals with respect to A are taken at the boundaries of the control volume. Leibniz's theorem states that for a fixed control volume, the volume is not a function of time and the time integral can be rewritten [14].

$$\frac{d}{dt} \int_V Q dV + \int_A F_j n_j dA - \int_A F_j^v n_j dA = \int_V S dV \quad (4.3)$$

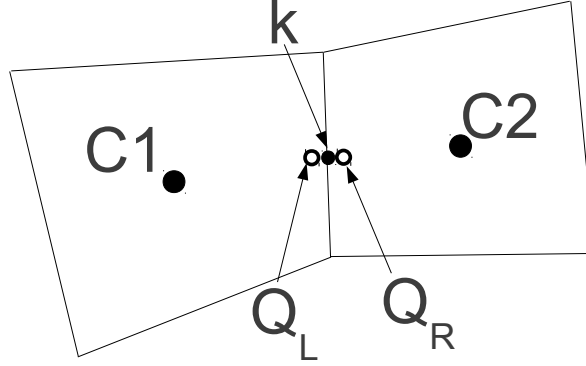


Fig. 4.1: A sample 2D mesh with cells c1 and c2. k represents the face between the cells.

To simplify the derivation, viscous terms are assumed to be negligible and the discussion is limited to the inviscid Euler equations. This yields the following equation:

$$\frac{d}{dt} \int_V Q dV + \int_A F_j n_j dA = 0 \quad (4.4)$$

The derivation thus far has made no discrete assumptions and has been entirely continuous. To discretize the above equation, the domain is split into cell-centered control volumes and Q is assumed to vary linearly in each cell. Q over the entire domain is piecewise linear, which creates a second-order-accurate scheme.

Discretely defining Q as linear has direct effects on the fluxes on the boundary. The fluxes can be exactly computed on the boundaries using only one quadrature point (mid-point rule). A sample of this configuration is found in Fig. 4.1. c1 and c2 are located at the center of the cells and k represents the quadrature point on the boundary. The inviscid fluxes can be written in the form below:

$$\int_A F_j n_j dA = \sum_k \int_{A_k} F_j n_j dA_k \approx \sum_k \hat{\mathcal{F}}_k = R(Q) \quad (4.5)$$

$\hat{\mathcal{F}}_k$ represents the numeric flux at the quadrature points at the cell interface and R represents the “residual.” \mathcal{F} is the area weighted flux given below:

$$\mathcal{F} = F_j n_j A \quad (4.6)$$

We now return to Eq 4.3 to apply this formulation to include viscous and source terms. Changing notion slightly, we now reside in cell o with weighted area fluxes between i cells around it in pseudo-time, yielding:

$$\frac{dQ_o}{d\tau} V_o + \sum_i \hat{\mathcal{F}}_{oi} - \sum_i \hat{\mathcal{F}}_{oi}^v = S_o V_o \quad (4.7)$$

where V_o is the volume of the current cell o . We simplify further in manner similar to the above method, but now including the viscous and source terms:

$$R_o(Q) = \sum_i \hat{\mathcal{F}}_{oi} - \sum_i \hat{\mathcal{F}}_{oi}^v - S_o V_o \quad (4.8)$$

thus getting:

$$\frac{dQ_o}{d\tau} V_o + R_o(Q) = 0 \quad (4.9)$$

This gives a starting point to now apply a time-stepping scheme to the equation. Explicit schemes calculate the state of a system at a later time from the state of the system at the current time, while implicit schemes find a solution by solving an equation involving both the current state of the system and the later one. Mathematically, if $Y(t)$ is the current system state and $Y(t + \Delta t)$ is the state at the later time, where Δt is some small time step, then, for an explicit method:

$$Y(t + \Delta t) = F(Y(t)) \quad (4.10)$$

while for an implicit method one solves an equation:

$$G(Y(t), Y(t + \Delta t)) = 0 \quad (4.11)$$

in order to find $Y(t + \Delta t)$.

Using k as the current pseudo-time step, and $k + 1$ as the next pseudo-time step and applying an implicit scheme to Eq 4.9 returns:

$$V_o \frac{Q_o^{k+1} - Q_o^k}{\Delta \tau} + R_o(Q^{k+1}) = 0 \quad (4.12)$$

where:

$$R_o(Q^{k+1}) = R_o(Q^k) + \frac{\partial R^k}{\partial Q} (Q_o^{k+1} - Q_o^k) \quad (4.13)$$

applying Eq 4.13 to Eq 4.12 and letting $\Delta Q^k = Q_o^{k+1} - Q_o^k$ yields:

$$\left[\frac{V_o}{\Delta \tau} I + \frac{\partial R^k}{\partial Q} \right] (\Delta Q^k) = -R_o(Q^k) \quad (4.14)$$

where I is the identity matrix.

Eq 4.14 is the general form that the strand solver uses. It is in this equation that we shall refer to the RHS and LHS from here on out. From observation, it is clear that this form requires the calculation of the residual at k time step (RHS), is common to both explicit and implicit. This form also requires the calculation of the derivative of R^k with respect to Q (referred to as LHS generally), which is unique to an implicit scheme. The next section describes the methods used to calculate the residual $R_o(Q)$ and the residual Jacobian $\frac{\partial R^k}{\partial Q}$.

4.2 Discretization of FANS and SA Turbulence Model System

This section directly follows from the previous and describes the necessary discretization methods required for the strand solver in an implicit scheme. We shall break $R_o(Q^k)$ back into its original form of $\sum_i \hat{\mathcal{F}}_{oi}$, $\sum_i \hat{\mathcal{F}}_{oi}^v$ and $S_o V_o$

4.2.1 Inviscid Flux

Returning to the notation formed in Eq 4.4 and 4.5, $\hat{\mathcal{F}}_k$ is defined by the following:

$$\hat{\mathcal{F}}_k = \frac{1}{2} (\mathcal{F}(Q_L) + \mathcal{F}(Q_R)) - D_k(Q_R, Q_L) \quad (4.15)$$

where $D_k(Q_R, Q_L)$ is the artificial dissipation, Q_L and Q_R represent the reconstructed terms.

The stability of Eq. 4.15 is dependent on the choice of the artificial dissipation term. Without dissipation, the numerical flux becomes immediately unstable. Adding a proper dissipation model makes Eq. 4.15 local extremum diminishing (LED). LED schemes require that local maxima decrease and local minima increase. LED schemes ensure that the local extrema are bounded and prevent them from diverging.

Our attention now turns to calculating the inviscid part of $\frac{\partial R^k}{\partial Q}$. To calculate the LHS of inviscid FANS-SA system of equations, we let:

$$|A| = \left| \frac{\partial \mathcal{F}}{\partial Q} \right| = R |\Lambda| R^{-1} \quad (4.16)$$

where Λ are the known eigenvalues defined as:

$$\Lambda = \text{diag}(u', u', u' + c, u' - c, u') \quad (4.17)$$

We define c as the speed of sound, and $u' = u_n - v_n$. We now have u_n defined as:

$$u_n = \frac{A_x u + A_y v}{|A|} = n_x u + n_y v \quad (4.18)$$

Knowing Λ and A , we can solve for R and R^{-1} .

4.2.2 Viscous Flux

The viscous portion of R_o^k and $\frac{\partial R^k}{\partial Q}$ must now must be calculated. We define the weighted area viscous flux \mathcal{F}^v as:

$$\mathcal{F}^v = \overline{A} \cdot \overline{F}^v = A_x F^v + A_y G^v = B_{xf}^v \frac{\partial Q^P}{\partial x} + B_{yf}^v \frac{\partial Q^P}{\partial y} \quad (4.19)$$

B_{xf}^v and B_{yf}^v are matrices containing the constants that lie in front of their respective differentiated primitive variables. This form now provides a format to solve the viscous portion of the residual.

$$B_{xf}^v = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{4}{3}A_x\mu & A_y\mu & 0 & 0 \\ 0 & -\frac{2}{3}A_y\mu & A_x\mu & 0 & 0 \\ 0 & 2\mu(A_x u - \frac{q_n}{3}) & \mu(A_x v + A_y u) & A_x k & 0 \\ 0 & 0 & 0 & 0 & A_x \frac{\eta}{\sigma} \end{bmatrix} \quad (4.20)$$

$$B_{yf}^v = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & A_y\mu & -\frac{2}{3}A_x & 0 & 0 \\ 0 & A_x\mu & \frac{4}{3}A_y\mu & 0 & 0 \\ 0 & \mu(A_y u + A_x v) & 2\mu(A_y v - \frac{q_n}{3}) & A_y k & 0 \\ 0 & 0 & 0 & 0 & A_y \frac{\eta}{\sigma} \end{bmatrix} \quad (4.21)$$

The viscous flux Jacobian needs to now be addressed to satisfy the residual Jacobian. We wish to calculate the differential of the viscous flux with respect to Q and also with respect to the gradient of Q.

$$\text{at a face}(f) : \mathcal{F}^v = \mathcal{F}^v(Q_f, Q_{xf}^P, Q_{yf}^P) \quad (4.22)$$

Thus we have:

$$\frac{\partial \mathcal{F}^v}{\partial Q} = \frac{\partial \mathcal{F}^v}{\partial Q_f} \frac{\partial Q_f}{\partial Q} + \frac{\partial \mathcal{F}^v}{\partial Q_{xf}^P} \frac{\partial Q_{xf}^P}{\partial Q^P} \frac{\partial Q^P}{\partial Q} + \frac{\partial \mathcal{F}^v}{\partial Q_{yf}^P} \frac{\partial Q_{yf}^P}{\partial Q^P} \frac{\partial Q^P}{\partial Q} \quad (4.23)$$

We can simplify this greatly if we let $\frac{\partial Q^P}{\partial Q} = \Gamma^P$, $\frac{\partial Q_{xf}^P}{\partial Q^P} = B_{xf}^v$, $\frac{\partial Q_{yf}^P}{\partial Q^P} = B_{yf}^v$ and $\frac{\partial \mathcal{F}^v}{\partial Q_f} = C_f^v$.

Recall that B_{xf}^v and B_{yf}^v were defined previously.

$$\frac{\partial \mathcal{F}^v}{\partial Q} = C_f^v \frac{\partial Q_f}{\partial Q} + [B_{xf}^v \frac{\partial Q_{xf}^P}{\partial Q^P} + B_{yf}^v \frac{\partial Q_{yf}^P}{\partial Q^P}] \Gamma^P \quad (4.24)$$

The face reconstruction terms must then be linearized. If we go from node a to b through a face with nodes 1 and 2 on each corner we may define the linearization of the reconstruction terms on fine levels as:

$$\begin{pmatrix} Q_{xf}^P \\ Q_{yf}^P \end{pmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{pmatrix} A_x \Delta Q_{12}^P \\ A_y \Delta Q_{ab}^P \end{pmatrix} \quad (4.25)$$

Choose $Q_f = \frac{1}{2}(Q_a + Q_b)$, thus we may generalize the definition as:

$$\begin{pmatrix} Q_{xf}^P \\ Q_{yf}^P \end{pmatrix} = \begin{pmatrix} D_x \Delta Q_{ab}^P \\ D_y \Delta Q_{ab}^P \end{pmatrix} \quad (4.26)$$

where D_x and D_y are the constants that change with respect to multi-grid level (either fine or course). Putting these changes together produces:

$$\frac{\partial \mathcal{F}^v}{\partial Q} = \frac{C_f^v}{2} (I_a + I_b) + B_f^v (\Gamma_b^P - \Gamma_a^P) \quad (4.27)$$

where:

$$B_f^v = D_x B_{xf}^v + D_y B_{yf}^v \quad (4.28)$$

A final form of the viscous Jacobian can be written as:

$$\frac{\partial \mathcal{F}^v}{\partial Q} = (\frac{C_f^v}{2} + B_f^v \Gamma^P) \dot{b} + (\frac{C_f^v}{2} - B_f^v \Gamma^P) \dot{a} \quad (4.29)$$

4.2.3 Source Terms

Calculating the residual portion of the source term is easy, as it is simply the source term at the cell multiplied by the volume of the cell. Calculating the residual Jacobian portion of the source term however is a complicated process. Similar to the viscous flux

Jacobian, we are required to calculate the differential of the source term with respect to Q and the gradient of Q .

$$S = P - D + T = P(Q, \nabla Q) - D(Q, \nabla Q) + T(Q, \nabla Q) \quad (4.30)$$

An example of how the source terms are linearized is shown below. Source terms below are differentiated with respect to Q ; $P(Q)$, $D(Q)$, and $T(Q)$. Note that all these derivatives have been greatly simplified, and more in-depth derivatives have not been presented.

$$\frac{\partial P}{\partial q_i} = \begin{cases} C_{b1}(\frac{\partial \tilde{S}}{\partial q_i} \rho \tilde{\nu} + \frac{\partial \rho \tilde{\nu}}{\partial q_i} \tilde{S})(1 - f_{t2}) + \rho \tilde{\nu} \tilde{S} C_{b1} \frac{\partial f_{t2}}{\partial q_i}, & \tilde{\nu} \geq 0 \\ C_{b1}(1 - C_{t3}) \Omega \frac{\partial \rho \tilde{\nu}}{\partial q_i}, & \tilde{\nu} < 0 \end{cases} \quad (4.31)$$

$$\frac{\partial D}{\partial q_i} = \begin{cases} \frac{C_{\omega 1}}{d^2} (\frac{\partial \rho \tilde{\nu}^2}{\partial q_i} f_w + \frac{\partial f_w}{\partial q_i} \rho \tilde{\nu}^2) - \frac{C_{\omega 1}}{\kappa^2 d^2} (\frac{\partial \rho \tilde{\nu}^2}{\partial q_i} f_{t2} + \frac{\partial f_{t2}}{\partial q_i} \rho \tilde{\nu}^2), & \tilde{\nu} \geq 0 \\ -\frac{C_{\omega 1}}{d^2} (\frac{\partial \rho \tilde{\nu}^2}{\partial q_i}), & \tilde{\nu} < 0 \end{cases} \quad (4.32)$$

$$\frac{\partial T}{\partial q_i} = \frac{C_{b2}}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \rho}{\partial q_i} - \frac{1}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \rho}{\partial x_k} (\frac{\partial \tilde{\nu}}{\partial q_i} f_n + \frac{\partial f_n}{\partial q_i} \tilde{\nu} + \frac{\partial \nu}{\partial q_i}) \quad (4.33)$$

Below we show the source terms with derivatives with respect to the gradient of Q ; $P(\nabla Q)$, $D(\nabla Q)$, and $T(\nabla Q)$.

$$\frac{\partial P}{\partial q_{xi}} = \begin{cases} \rho C_{b1}(1 - f_{t2}) \tilde{\nu} \frac{\partial \tilde{S}}{\partial q_{xi}}, & \tilde{\nu} \geq 0 \\ \rho C_{b1}(1 - C_{t3}) \tilde{\nu} \frac{\partial \Omega}{\partial q_{xi}}, & \tilde{\nu} < 0 \end{cases} \quad (4.34)$$

$$\frac{\partial D}{\partial q_{xi}} = \begin{cases} \rho C_{\omega 1} (\frac{\tilde{\nu}}{d})^2 \frac{\partial f_w}{\partial q_{xi}}, & \tilde{\nu} \geq 0 \\ 0, & \tilde{\nu} < 0 \end{cases} \quad (4.35)$$

$$\frac{\partial T}{\partial q_{xi}} = \frac{\rho C_{b2}}{\sigma} \frac{\partial}{\partial q_{xi}} (\frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \tilde{\nu}}{\partial x_k}) - \frac{(\nu + \tilde{\nu} f_n)}{\sigma} \frac{\partial}{\partial q_{xi}} (\frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \rho}{\partial x_k}) \quad (4.36)$$

4.3 Boundary Conditions and Boundary Discretization

Boundaries for strand grids are very similar to the internal discretization; however,

unlike internal discretization the flux on the wall is now the actual problem boundary instead of a neighboring cell. Strands use only a cell-centered structure. Since known values in cell-centered paradigms are only known at the centroid of the cell, implementing equations at the boundaries is difficult. To apply equations at the boundary, a “ghost-node” is placed at the quadrature point of the boundary cell (b). The values from the interior of the domain are extrapolated in a similar fashion to the interior reconstruction to the boundary (e). Fig. 4.2 shows a simple one dimensional example of this.

$$Q_e = Q_i + (x_e - x_i) \left. \frac{\partial Q}{\partial x} \right|_i \quad (4.37)$$

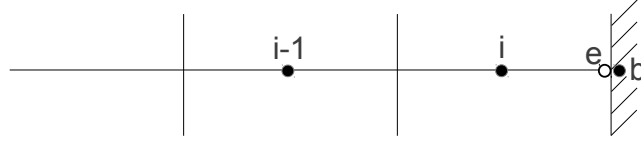


Fig. 4.2: A sample 1D cell centered boundary with extrapolated point and ghost node.

The general form of Dirichlet boundary conditions consists of $B(Q_s) = b$, where m number of equations must be satisfied for each condition. Allmaras [43] and Katz [35] describe this in detail further. The boundary fluxes can be presented in the form of a boundary residual, $R_b = B(Q_s) - b$, similar to interior fluxes. Here b is the constant values found on the boundary, and $B(Q_s)$ some auxiliary variables used for convenience, primarily the primitive variables.

In turbulent flow, the SA model requires [19,40] that the farfield and freestream boundary conditions be defined as:

$$\tilde{\nu}_{farfield} = (3 \rightarrow 5)\tilde{\nu}_{\infty} \quad (4.38)$$

and the wall boundary conditions be defined as:

$$\tilde{\nu}_{wall} = 0 \quad (4.39)$$

Combining the above SA boundary conditions with boundary conditions given by Katz [35], we may obtain the equations for the boundary residual for key boundary conditions; inflow, outflow, viscous wall, inviscid wall, and pure dirichlet where the entire state is specified and fixed. The *spec* values refer to specified user values. An example of an outflow that specifies a pressure has a boundary residual of:

$$R_b = \left\{ \begin{array}{c} p_b - p_{spec} \\ u - u_e \\ v - v_e \\ T_b - T_e \\ \tilde{\nu} - \tilde{\nu}_e \end{array} \right\} = 0 \quad (4.40)$$

An Inflow that specifies a pressure has a boundary residual of:

$$R_b = \left\{ \begin{array}{c} p_b - p_{spec} \\ u_t \\ u_n \\ T_b - T_e \\ \tilde{\nu}_b - \tilde{\nu}_{spec} \end{array} \right\} = 0 \quad (4.41)$$

Here, $u_t = \mathbf{u} \cdot \mathbf{t}$ is the velocity tangential to the inflow plane, along the tangential vector, $\mathbf{t} = (-ny, nx)$. In a similar fashion, $u_n = \mathbf{u} \cdot \mathbf{n}$ is the velocity normal to the inflow plane along the normal vector, $\mathbf{n} = (nx, ny)$. The viscous wall boundary residual is given as:

$$R_b = \begin{Bmatrix} p_b - p_e \\ u_b + u_e \\ v_b + v_e \\ T_b - T_e \\ \tilde{\nu}_b + \tilde{\nu}_e \end{Bmatrix} = 0 \quad (4.42)$$

An inviscid wall boundary residual is given as:

$$R_b = \begin{Bmatrix} \rho - \rho_e \\ u_t \\ u_n \\ \rho e - \rho e_e \\ \rho \tilde{\nu} - \rho \tilde{\nu}_e \end{Bmatrix} = 0 \quad (4.43)$$

The inviscid wall auxiliary variables Q_s use the standard conserved variables seen in the equations of motion instead of the primitive variables. The Dirichlet boundary residual is given as:

$$R_b = \begin{Bmatrix} p - p_{spec} \\ u - u_{spec} \\ v - v_{spec} \\ T - T_{spec} \\ \tilde{\nu} - \tilde{\nu}_{spec} \end{Bmatrix} = 0 \quad (4.44)$$

4.4 Method of Manufactured Solutions

To start the Method of Manufactured Solutions, we must first give values to the conserved variables. The values generally consist of some positive smooth trigonometric function. These terms for the conserved variables shall be called the source variables to avoid confusion. An example of how this is done for density is shown below:

$$\rho = B(1 + a_x \sin(b_x - c_x x) + a_y \sin(b_y - c_y y) + a_{xy} \sin(b_{xy} - c_{xy} xy)) \quad (4.45)$$

Here x and y represent Cartesian coordinates along the grid. B is some base value defined by the user to keep the values in the solution realistic. a_x , a_y , a_{xy} , b_x , b_y , b_{xy} , c_x , c_y , and c_{xy} all represent some arbitrary constants chosen to keep the function smooth and positive. This type of function is done for all conserved variables.

From 3.40 we show the equation used for density in the fully coupled FANS and SA equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = 0 \quad (4.46)$$

applying MMS to this equation in steady state ($\frac{\partial \rho}{\partial t} = 0$), we let this equation equal to some known source term, denoted by S_ρ .

$$\frac{\partial(\rho u_j)}{\partial x_j} = S_\rho \quad (4.47)$$

The source term is calculated directly by differentiating the source variables and applying the constants defined above. In the 2D system for ρ we declared above, we have:

$$\begin{aligned} S_\rho &= F_x + G_y \\ F_x &= \rho \frac{\partial u}{\partial x} + u \frac{\partial \rho}{\partial x} \\ G_y &= \rho \frac{\partial v}{\partial y} + v \frac{\partial \rho}{\partial y} \end{aligned} \quad (4.48)$$

So in this example, we shall differentiate our source variables u and ρ from Eq 4.45 with respect to x and y to find values for F_x , G_y , and ultimately S_ρ . With a value for S_ρ , we now feed the source variables into the strand solver. If the algorithms have been properly implemented, then:

$$\frac{\partial(\rho u_j)}{\partial x_j} - S_\rho = error \quad (4.49)$$

As the strand solver is not exact and makes some assumptions and truncations, there will be some small difference (*error*) between the source term and the solution. We perform this test for increasing grid resolutions to show that as the number of cells increase for a fixed mesh size, the accuracy of the solution improves by an order of two. The results of this test is shown in Chapter 5.

Chapter 5

Results and Discussion

This section will present and discuss the results from the verification and validation of the Spalart-Allmaras turbulence model.

5.1 Verification

First, we perform grid refinement studies using the method of manufactured solutions [22]. This study ensures to the extent possible that we have not made coding mistakes in the strand grid discretization.

Five levels of grid refinement are used for the study, with 32, 64, 128, 256, and 512 cells along each side of the square, respectively. The manufactured solution is chosen based on smooth trigonometric functions similar to previous work [44]. An example of how these functions were set up and applied to the strand solver was shown in Chapter 4.

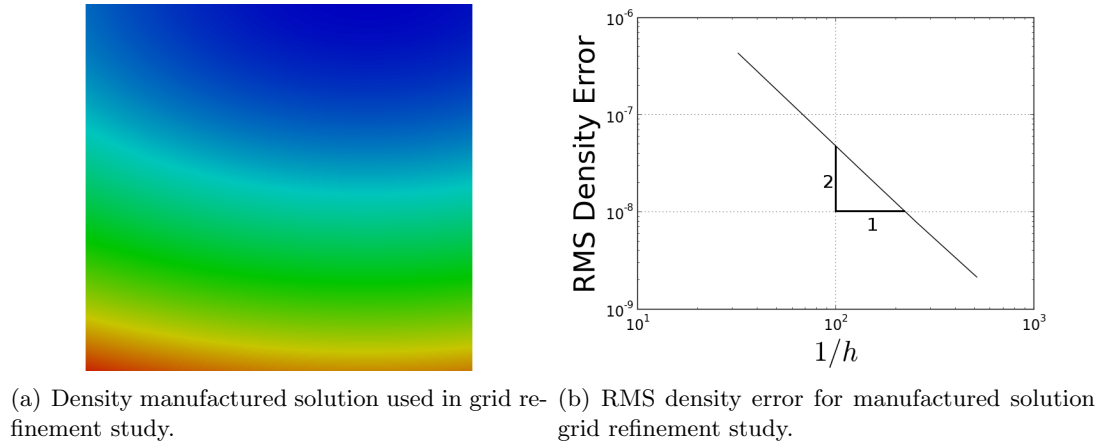


Fig. 5.1: MMS grid refinement study results.

The manufactured solution for density is shown in Figure 5.1(a). Dirichlet conditions enforcing the exact solution are used at all boundaries for this study. The results of the grid refinement study for isotropic strand distributions are shown in Figure 5.1(b). h is defined as the characteristic cell size; the strand length (1), over the number of cells per strand. Second order convergence was achieved for all conserved variables.

Work et al. [12] performed grid refinement studies using MMS on multi-strand and strand smoothing for laminar flow, allowing for any sensitivities of the flow solver to the different meshing strategies to be identified. MMS was set up in the same manner as it was in this work. Three sharp corner treatments were investigated: no smoothing or multi-strands (Figure 5.2(a)), extremely smoothed strands (Figure 5.2(b)), and multi-strands (Figure 5.2(c)). We perform the refinement study for both isotropic (Figure 5.3(a)) strand distributions, as well as anisotropic (Figure 5.3(b)) strand distributions for which the wall spacing is chosen to produce an aspect ratio of 10^3 for the first cell off the wall.

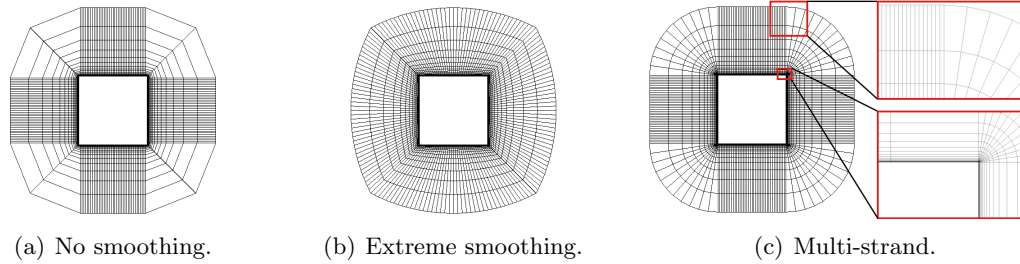
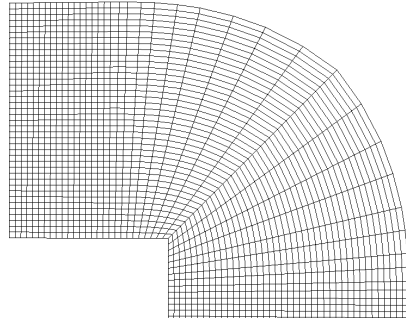
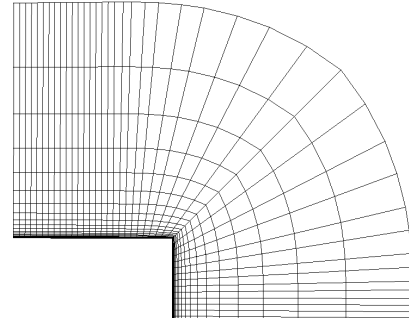
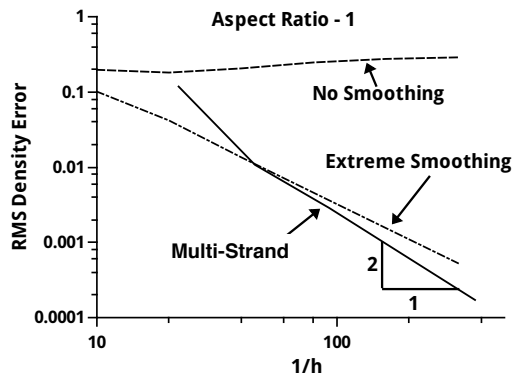


Fig. 5.2: Various meshing strategies used for manufactured solution grid refinement studies.

The results of the grid refinement study for isotropic and anisotropic strand distributions are shown in Figure 5.3. For both isotropic and anisotropic distributions, no special corner treatment (labeled “No Smoothing”) results in loss of error convergence. This is expected because without smoothing, the grid cells are not reduced in size properly as the grid is refined. Clearly, special treatment is required at sharp corners to achieve the design order of accuracy.



(a) Aspect ratio of 1 - grid.

(b) Aspect ratio of 10^3 - grid.

(c) Aspect ratio of 1 - error.

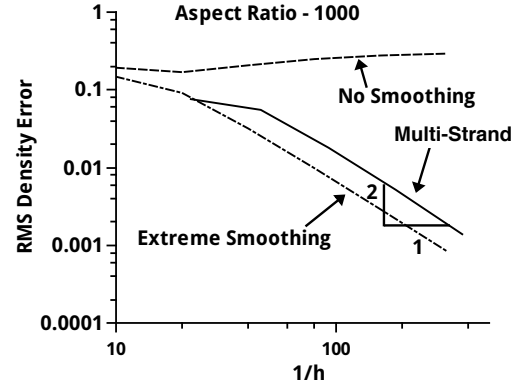
(d) Aspect ratio of 10^3 - error.

Fig. 5.3: RMS density error for manufactured solution grid refinement studies using various meshing strategies on isotropic and stretched strand distributions.

Second order accuracy is obtained for both smoothing and multi-strand approaches, for both isotropic and anisotropic strand distributions. The multi-strand approach gives slightly lower errors with isotropic strand distributions, perhaps due to reduced skewing of the grid cells. But for anisotropic high aspect ratio strand distributions, which would be used for turbulent RANS or FANS calculations, smoothed strands produce lower error than the multi-strand grids by about a factor of five. The explanation for this likely lies in the interaction of multi-strands and stretched node distributions, as shown in the red boxes of Figure 5.2(c). Multi-strands tend to produce highly discontinuous mesh spacing at different orientations, both at the strand ends, as well as the surface. Attempts were made to cluster surface points near the corners to avoid the thin high aspect ratio cells bordering the tiny wedge cells at the sharp corner. However, in doing this, the mesh discontinuity at the strand ends became more severe. It appears that eliminating these mesh discontinuities could prove difficult which is unfortunate since they seem to be responsible for higher errors as well as some convergence issues.

5.2 Validation

Extensive use of the NASA Langley turbulence modeling resource has been made for these cases. This resource is comprised of previous experimental results, theoretical results, and two independent compressible CFD codes; FUN3D and CFL3D [3].

5.2.1 Flat Plate with Zero Pressure Gradient

MMS allows for the solver to be verified on a mathematical level, however, there may still be problems with the code when running a real case. Thus additional verification is required. The turbulent flat plate validation case may be considered as verification through validation due to its simplicity.

The turbulent flat plate case was run at $M = 0.2$, at a Reynolds number of $Re = 5 \times 10^6$ based on length “1” of the grid. The body reference length is 2 units. The following plot shows the layout of the simple flat plate grids used for this study, along with the boundary conditions. An outflow BC was used instead of a farfield Riemann BC.

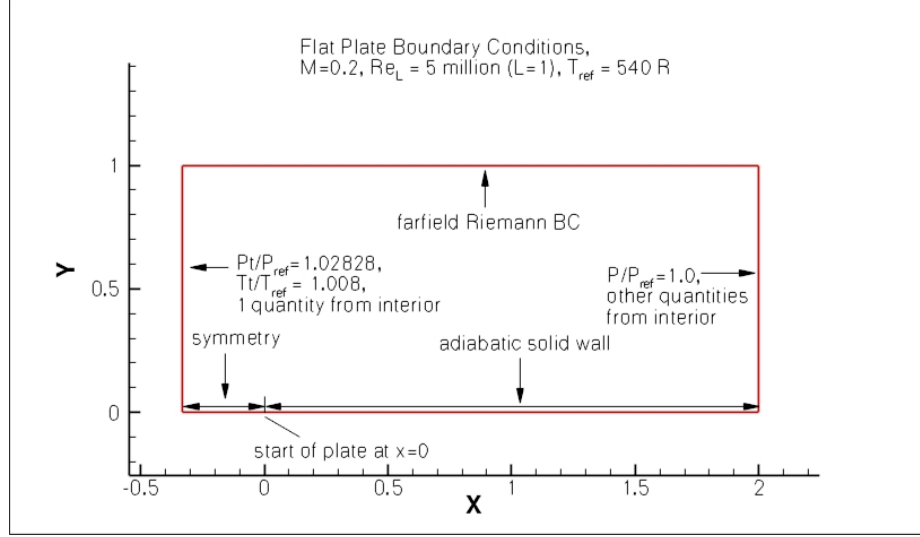


Fig. 5.4: Flat Plate Dimensions and Boundary Conditions [3]

Certain inputs and slight deviations were required to each model in order to comply with the NASA Langley turbulence resource validation case. The SA model only required that $\tilde{\nu}_{farfield} = 3\nu_{\infty}$.

The grid used for the flat plate validation is the 137×97 grid shown in Figure 5.5(a). The plate leading edge begins at $x = 0$ and extends for a length of two. A short inviscid wall entry way beginning at $x = -0.33$ is provided to allow for proper inflow conditions. Stagnation temperature and pressure are specified at the inflow, and static pressure is specified at the outflow. The turbulent viscosity field for this case is shown in Figure 5.5(b) which has been scaled by a factor of 40 vertically to facilitate visualization. Streamwise velocity and turbulent viscosity profiles are shown for two locations downstream on the plate, and are over plotted with FUN3D and CFL3D results in Figure 5.6.

Note that good agreement is obtained, even for the 137×97 grid, which is 16 times more coarse than the FUN3D and CFL3D results shown in the figure. The computed drag coefficient, which is entirely due to skin friction for this case, is shown in Table 5.1, along with FUN3D and CFL3D results for the same grid. The drag coefficient falls within the range predicted by the established codes.

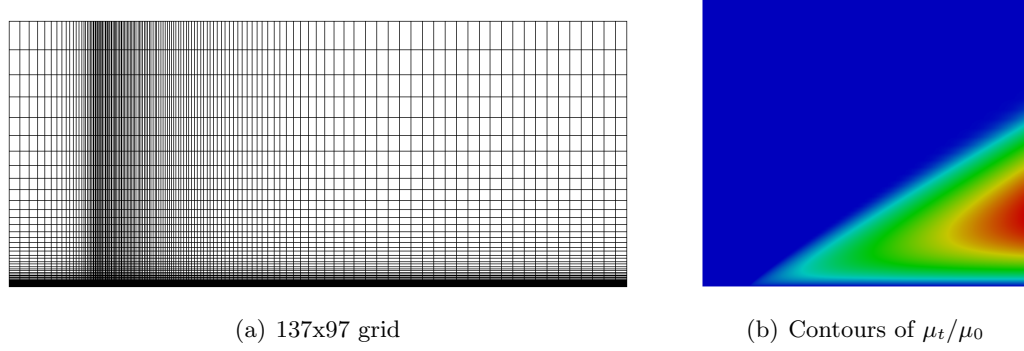


Fig. 5.5: Grid and turbulent viscosity contours for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$.

The $\rho\tilde{\nu}$ field for this case is shown in Figure 5.7 which has been scaled by a factor of 40 vertically to facilitate visualization. It is clear that $\tilde{\nu}$ reaches a negative value in the flow field. This plot confirms the need for the “negative” model.

5.2.2 Bump-in-channel

The bump-in-channel case is different from the simpler flat plate verification case because it involves wall curvature and, as a result, pressure gradients. It was run at a Mach number of $M = 0.2$, at a Reynolds number of $Re = 3 \times 10^6$ based on length “1” of the grid. The body reference length is 1.5 units. The lower wall is a viscous-wall bump extending from $x = 0$ to 1.5. The maximum bump height is $Y_0 = 0.05$. The exact definition of the bump is:

Table 5.1: Comparison of computed drag coefficients for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$.

	C_d
Strand	2.82287E-3
FUN3D (quads)	2.84005E-3
FUN3D (triangles)	2.80289E-3
CFL3D	2.86621E-3

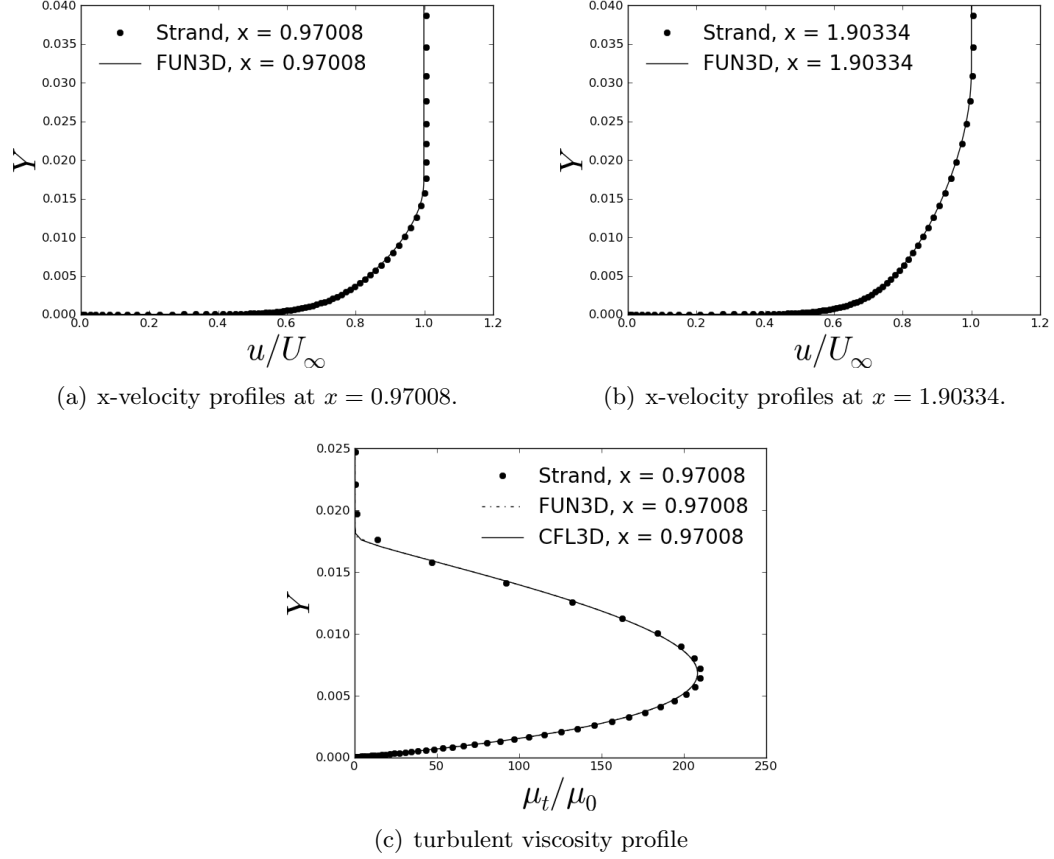


Fig. 5.6: Comparison of streamwise velocity and turbulent viscosity profiles for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$.

$$y = \begin{cases} 0.05(\sin(\pi \frac{x}{0.9} - (\frac{\pi}{3})))^4, & 0.3 \leq x \leq 1.2 \\ 0, & 0 \leq x < 0.3 \text{ and } 1.2 < x \leq 1.5 \end{cases} \quad (5.1)$$

The upstream and downstream farfield extends 25 units from the viscous-wall, with inviscid plane boundary conditions imposed on the lower wall between the farfield and the solid wall. The upper boundary is a distance of $y = 5.0$ high. It is taken to be an inviscid plane. Figure 5.8 shows the layout of this case, along with the boundary conditions.

The grid used for the bump-in-channel validation is the 353×161 grid shown in Figure 5.9(a) and a close up is shown in Figure 5.9(b). Figures 5.9 and 5.9(c) are scaled by a factor of 100 vertically and a factor of 16.667 horizontally to facilitate visualization. To

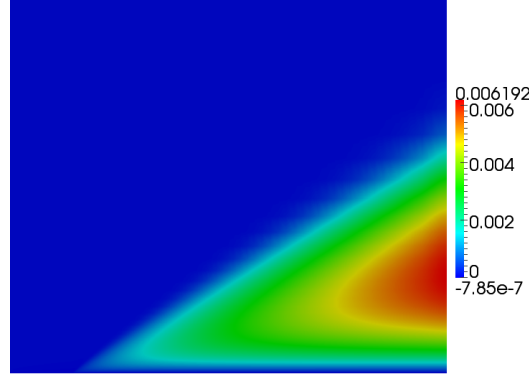
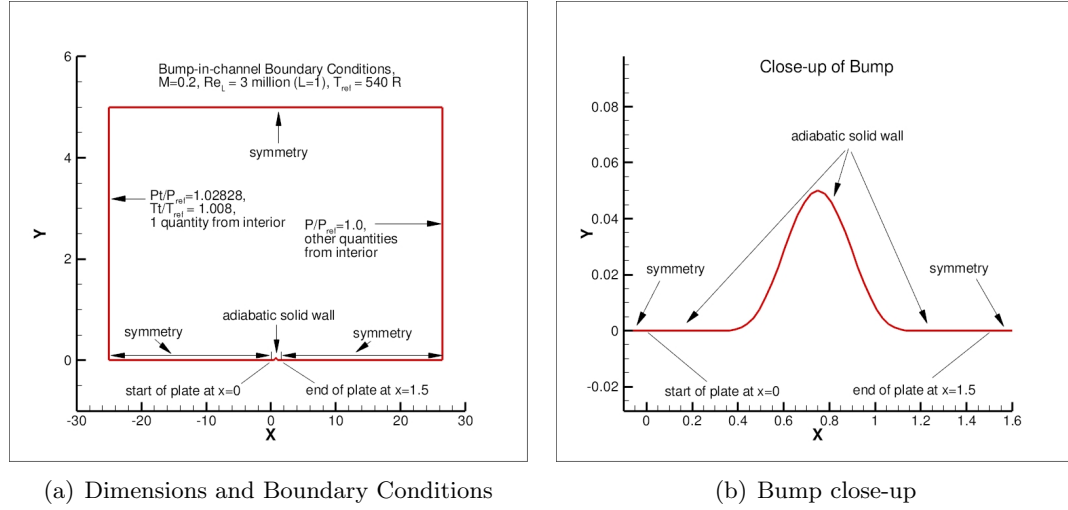
Fig. 5.7: Contours of $\rho \tilde{\nu}$ 

Fig. 5.8: Bump-in-channel Dimensions and Boundary Conditions [3]

avoid the internal corner issues over the bump, strands were set vertically only, and were not smoothed or skewed. Stagnation temperature and pressure are specified at the inflow, and static pressure is specified at the outflow, however similar input conditions will yield similar results. The turbulent viscosity field for this case is shown in Figure 5.9(c). Streamwise velocity and turbulent viscosity profiles are shown for two locations downstream on the bump, and are over plotted with FUN3D and CFL3D results in Figure 5.10.

Note that good agreement is obtained for all profiles, even for the 353×161 grid, which is 16 times more coarse than the FUN3D and CFL3D results shown in the figure. Below a

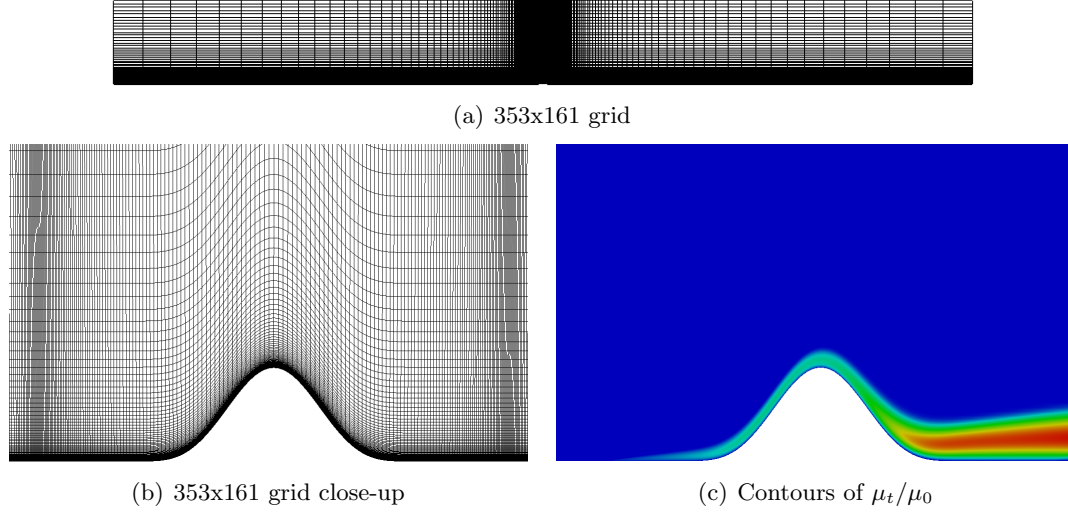
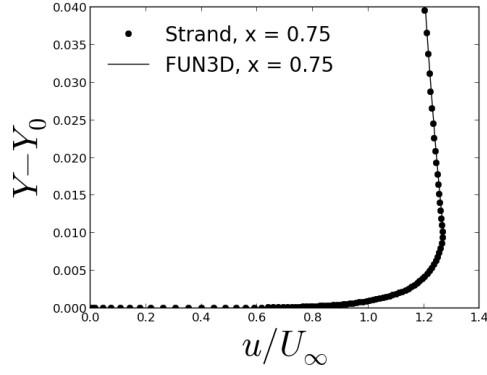
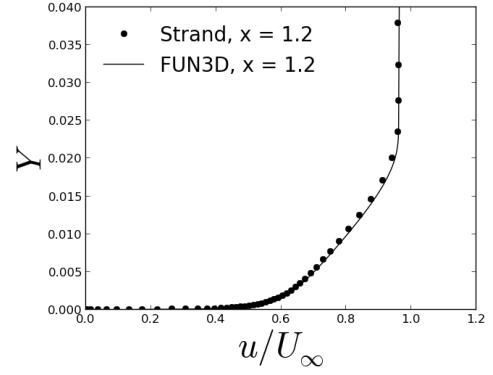
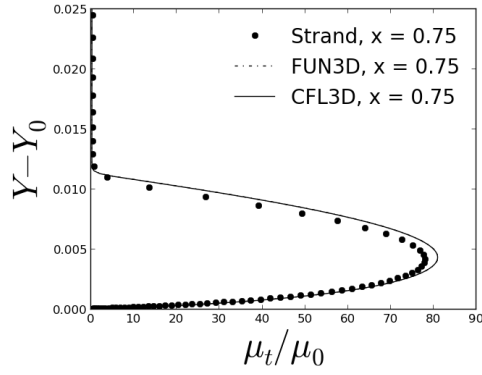


Fig. 5.9: Grid and turbulent viscosity contours for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$.

plot of the surface coefficient of pressure and friction along the bump is shown, and is over plotted with CFL3D and FUN3D results in Figure 5.11.

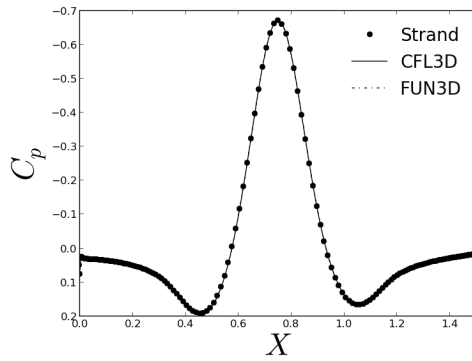
The NASA Langley turbulence resource [3] makes some useful observations about the surface skin friction. In this bump case the surface skin friction is singular (tends toward infinity) at the leading edge. The finer the grid, the more nearly singular the local behavior on a finite grid. There is also locally anomalous behavior in C_f at the back end of the bump wall (at $x = 1.5$), as is often seen in CFD solutions near trailing edges [45]. Figure 5.11 shows the local anomalous behavior that exists near the leading edge ($x = 0$) due to singular behavior of the solution, as well as near the trailing edge ($x = 1.5$) due to numerical influences. These behaviors differ for the three codes, and result in small local deviations that can be seen when zoomed into the two locations. All three codes are seen to yield nearly identical results over most of the bump wall.

The computed drag coefficient, is shown in Table 5.2, along with FUN3D and CFL3D results for the same grid. The drag coefficient falls within the range predicted by the established codes. It must be emphasized again that the C_d results in Table 5.2 obtained by FUN3D and CFL3D were run on very fine grids; while the strand code ran on a grid

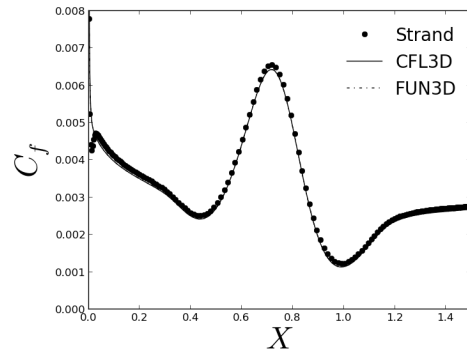
(a) x-velocity profiles at $x = 0.75$.(b) x-velocity profiles at $x = 1.2$.

(c) turbulent viscosity profile

Fig. 5.10: Comparison of streamwise velocity and turbulent viscosity profiles for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$.



(a) Surface coefficient of pressure



(b) Surface coefficient of friction

Fig. 5.11: Bump-in-channel surface coefficient of pressure and friction

16 times coarser. In the NASA Langley turbulence resource, CFL3D and FUN3D obtained higher values of C_d on coarse grids. As the number of cells in the grid increased, both codes tended asymptotically towards the numbers given in Table 5.2.

5.2.3 NACA 0012 Airfoil

With the SA model verified for the flat plate and bump-in-channel cases, we may validate the flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$ at various angles of attack. To reduce the effect from the farfield boundaries, the Cartesian grid is extended 500 chords in all directions. Riemann invariant boundary conditions are imposed at the farfield. Portions of the Cartesian grid system are shown in Figure 5.12. A baseline grid consisting of 10 levels of refinement and 16,348 cells is used, shown in Figure 5.12(a).

Additionally, a wake-capturing Cartesian grid consisting of 15 levels of refinement and 304,348 cells was used, shown in Figure 5.12(b). The wake capturing grid is made quite wide and fine to handle high angles of attack. The grid count could be significantly reduced by using a high-order method and automatic grid adaptation, though these avenues are not pursued. Nonetheless, the wake-capturing Cartesian grid is still quite efficient, due to the efficiency of the block-structured Cartesian grid code.

The strand grid for this case consists of a NACA 0012 airfoil containing 320 surface nodes and 64 cells along each strand, yielding 20,480 total strand cells. Both multi-strands and strand smoothing are tested, with and without the wake-refined Cartesian grid. The four mesh combinations are shown in Figure 5.13. The wake-refined Cartesian grids are wider than would normally be needed and could be tailored to specific cases through more sophisticated Cartesian mesh adaptation strategies.

Table 5.2: Comparison of computed drag coefficients for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$.

	C_d
Strand	3.57559E-3
FUN3D	3.56106E-3
CFL3D	3.57238E-3

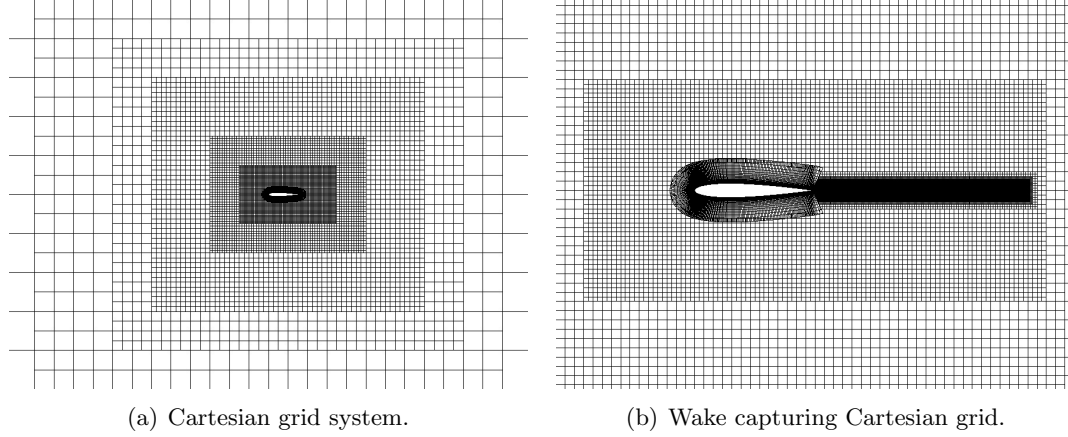


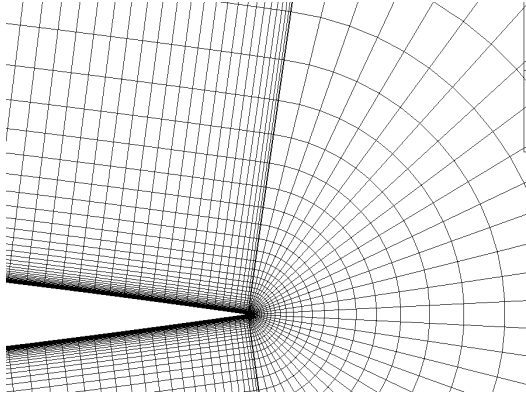
Fig. 5.12: Grid system for flow over a NACA 0012 airfoil.

The surface pressure coefficients for $\alpha = 0^\circ, 10^\circ, 15^\circ$ are shown in Figure 5.14, and compared with the experimental data of Gregory and O'Reilly [46]. The Gregory data is actually taken at $Re = 3 \times 10^6$, not $Re = 6 \times 10^6$, but little change in pressure and lift is observed between the two Reynolds numbers. The figure only displays the results for multi-strand treatment without the wake-refined Cartesian grid because all methods matched the pressure data reasonably well.

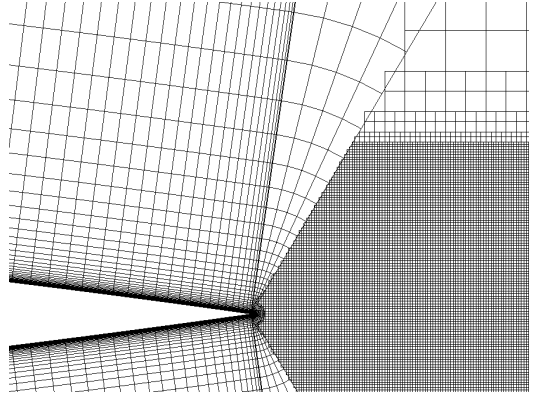
Greater discrepancies in the methods are uncovered by observing computed lift, and drag values. Figures 5.15 and 5.16 show lift coefficient versus angle of attack and drag coefficient versus lift coefficient, respectively, along with the corresponding experimental data of Ladson [47]. The lift data is matched reasonably well by most mesh configurations, with

Table 5.3: Comparison of computed lift and drag coefficients for flow over a NACA 0012 airfoil at $M = 0.15$, $\alpha = 15^\circ$ $Re = 6 \times 10^6$. Note that the FUN3D and CFL3D results use a much finer grid.

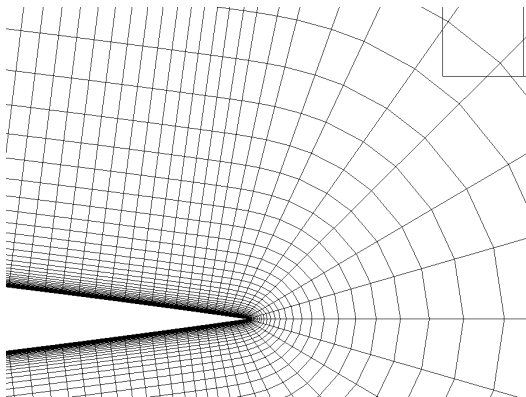
	C_l	C_d
Multi-strand, no Cart. refine (320 points)	1.5587	0.02332
Multi-strand, Cart. refine (320 points)	1.5457	0.02284
Smoothed strand, no Cart. refine (320 points)	1.6127	0.02417
Smoothed strand, Cart. refine (320 points)	1.5578	0.02189
FUN3D (513 points)	1.5547	0.02159
CFL3D (513 points)	1.5461	0.02124



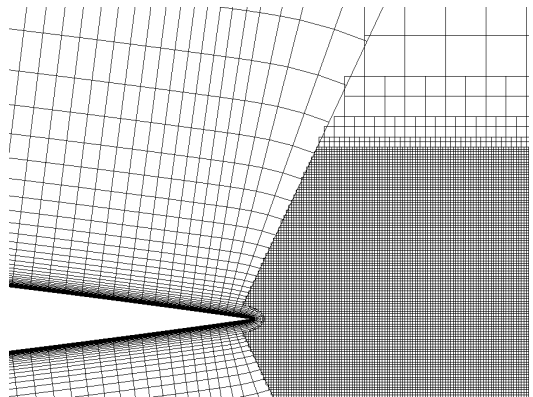
(a) Multi-strands, no wake grid.



(b) Multi-strands, wake grid.



(c) Smoothed strands, no wake grid.



(d) Smoothed strands, wake grid.

Fig. 5.13: Various meshing strategies for the trailing edge of a NACA 0012 airfoil.

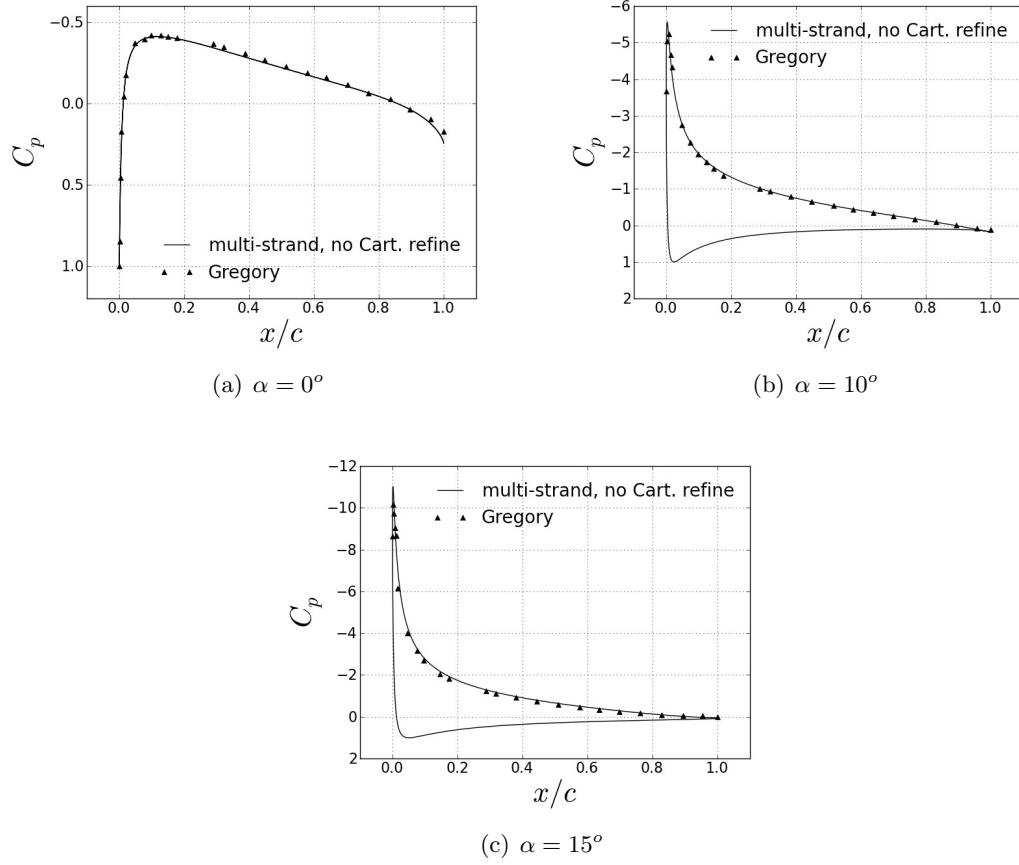


Fig. 5.14: Pressure coefficient for flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$ at various angles of attack. Only multi-strand results with no wake Cartesian refinement are shown. Other meshes were indistinguishable.

the exception of the smoothed strand case with no Cartesian refinement, which significantly over predicts lift for $\alpha = 15^\circ$. This case also produces the most error in the drag for all angles of attack. Importantly, the smoothed strands actually match the data better than any other configuration with the use of the wake-refined Cartesian grid. The multi-strand cases improve less in the presence of the wake-refined Cartesian grid. For clarity, the drag results for $\alpha = 15^\circ$, shown in Figure 5.16, are summarized in Table 5.3, along with FUN3D and CFL3D results using a much finer grid (513 surface nodes instead of 320). Again, it is clear that the smoothed strands produce the worst results of any case without the wake-refined grid, but the best results of any case with the wake-refined grid. The multi-strand

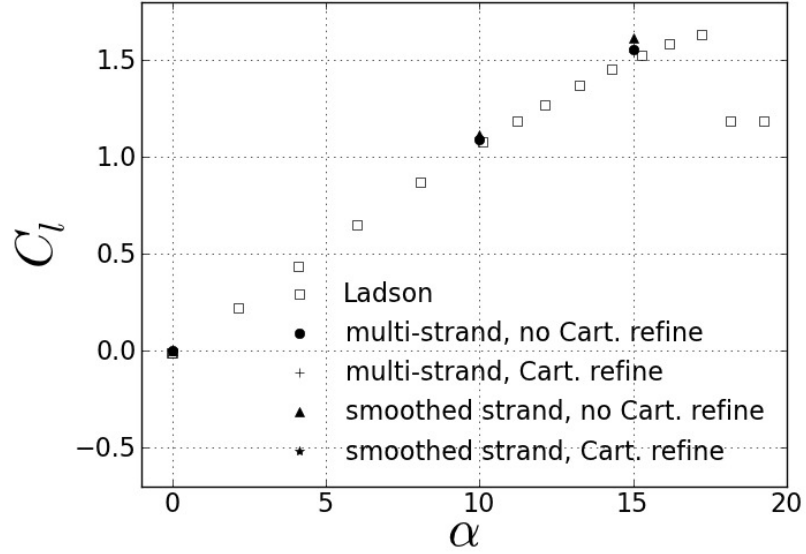


Fig. 5.15: C_l vs. α compared to experiment using various meshing strategies for flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$.

configurations lie in the middle and are less affected by the wake-refined Cartesian grid.

The above observations are explained by examining the ability of the various mesh configurations to accurately resolve the thin wake leaving the airfoil trailing edge. Resolution of the wake is critical for accurately computing lift, and especially drag, for this case. Field plots of velocity magnitude show the ability of the strand method to accurately capture the wake for various angles of attack in Figure 5.17. The figures on the left use smoothed strands without the wake-refined Cartesian grids, while the figures on the right use smoothed strands with the wake-refined grids. Clearly, the refined Cartesian grids capture the wake much better, which explains the improved drag results in Table 5.3. But there is a more subtle, yet critical observation beyond this. In the multi-strand case, the increased refinement at the trailing edge provided by the multi-strands provides reasonable resolution of the wake, even without the refined Cartesian grid. In the smoothed case, however, the resolution is much worse at the trailing edge than is the case with multi-strands, which explains why

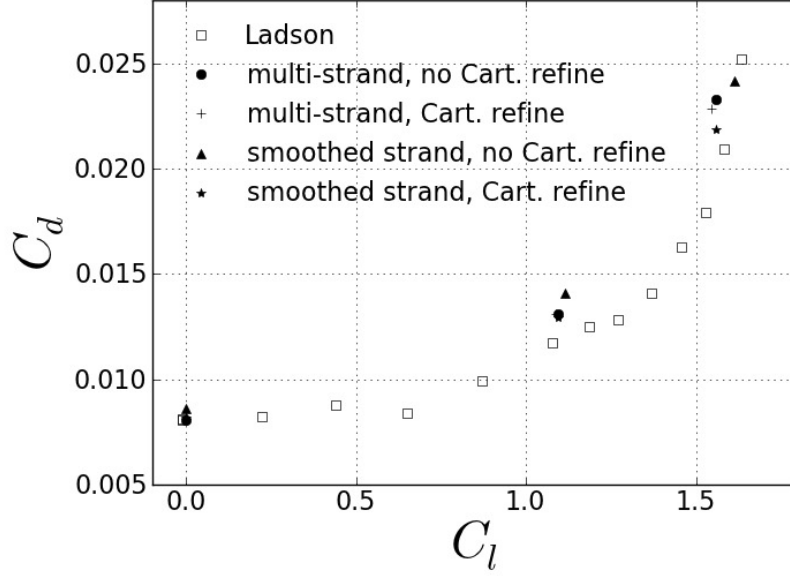


Fig. 5.16: C_l vs. C_d compared to experiment using various meshing strategies for flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$.

smoothed strands are more reliant on the Cartesian grids to obtain accurate results. Also notable are the lack of mesh discontinuities in the smoothed grid which leads to more accurate results than the multi-strand approach.

In summary, the optimal mesh configuration for this case appears to be smoothed strands with wake-refined Cartesian grids. Use of the smoothed strands avoids mesh discontinuities present with multi-strands, although the accuracy of smoothed strands is highly reliant on the availability of Cartesian refinement for proper resolution of the wake and accurate drag computation. The use of a high-order Cartesian method, such as in the Helios [48], is expected to dramatically improve wake resolution even further, as well as resolve other important features, such as vortical shedding.

5.2.4 NACA 4412 Trailing Edge Separation

With the simpler NACA 0012 airfoil case validated, a more complex airfoil, in the form

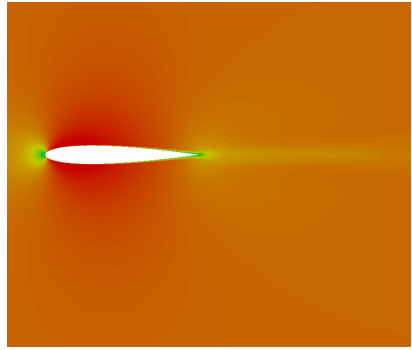
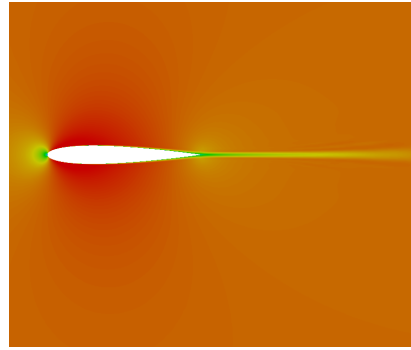
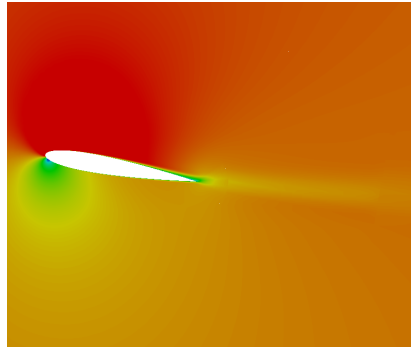
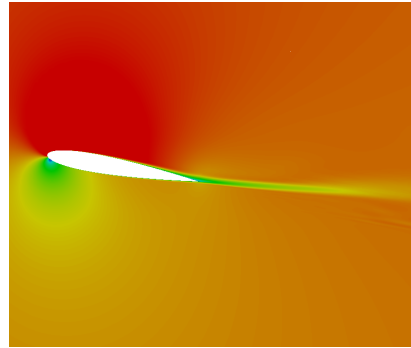
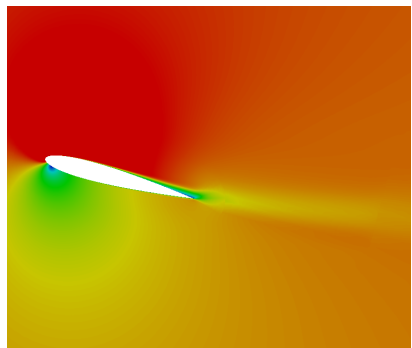
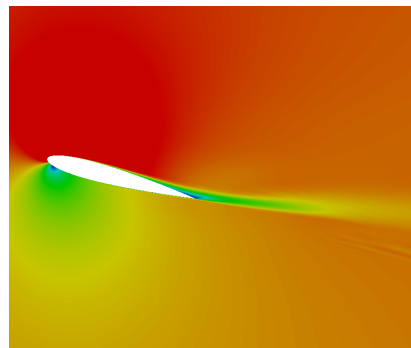
(a) no wake grid, $\alpha = 0^\circ$ (b) wake grid, $\alpha = 0^\circ$ (c) no wake grid, $\alpha = 10^\circ$ (d) wake grid, $\alpha = 10^\circ$ (e) no wake grid, $\alpha = 15^\circ$ (f) wake grid, $\alpha = 15^\circ$

Fig. 5.17: Field plots of velocity magnitude for various α , showing the effect of Cartesian refinement in the wake region. All cases use smoothed strands.

of the NACA 4412 airfoil, can now be tested. This case aims to study the flow separation at the trailing edge of the airfoil and flow characteristics at the wake.

For purposes of the validation, the definition of the airfoil shape is slightly altered so that the airfoil closes at chord $x = 1$ with a sharp trailing edge.

Experimental test data is provided by Coles and Wadcock [49, 50] Flowfield characteristics were measured with a flying hot-wire for the airfoil at 13.87 degrees angle of attack. A Mach number of $M = 0.09$ at a Reynolds number of $Re = 1.52 \times 10^6$ per airfoil chord was used. Both the upper and lower boundary layers were tripped in the experiment (2.5c on the upper surface and 10.3c on the lower surface). Like the other validation cases performed so far, FUN3D and CFL3D will be used for comparative purposes. Note that FUN3D and CFL3D have grids with a farfield outer boundary extending to 100c, but the experiment was in a relatively small wind tunnel, which likely had some influence. The NASA-Langley turbulence resource only provides the results data from CFL3D as the FUN3D code had almost identical results.

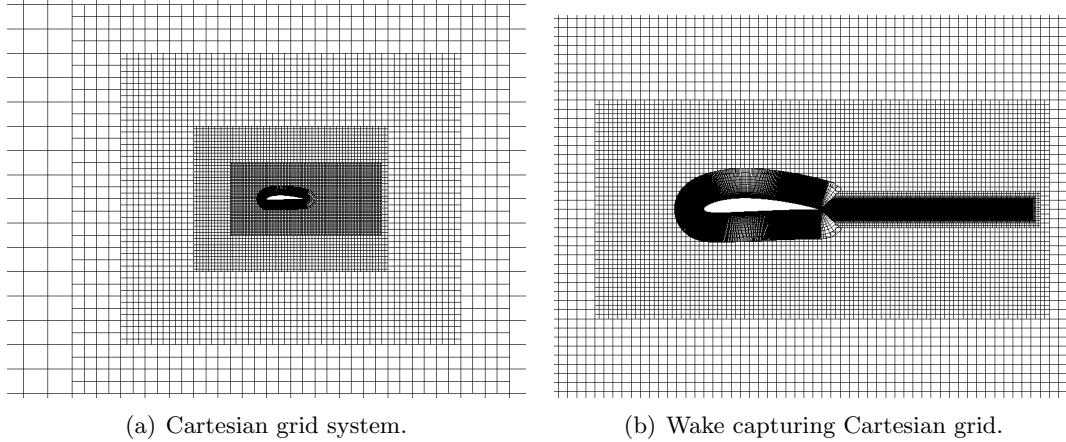


Fig. 5.18: Grid system for flow over a NACA 4412 airfoil.

The strand-Cartesian solver extends the Cartesian grid 500 chords in all direction to reduce the effect from the farfield boundaries. Riemann invariant boundary conditions are

imposed at the farfield. Portions of the Cartesian grid system are shown in Figure 5.18. A baseline grid consisting of 10 levels of refinement and 12,876 cells is used, shown in Figure 5.18(a). Additionally, a wake-capturing Cartesian grid consisting of 15 levels of refinement and 411,908 cells was used, shown in Figure 5.18(b). The wake refining grid was made quite wide to handle the high angle of attack. The total number of cells needed could be reduced with Cartesian grid adaption strategies.

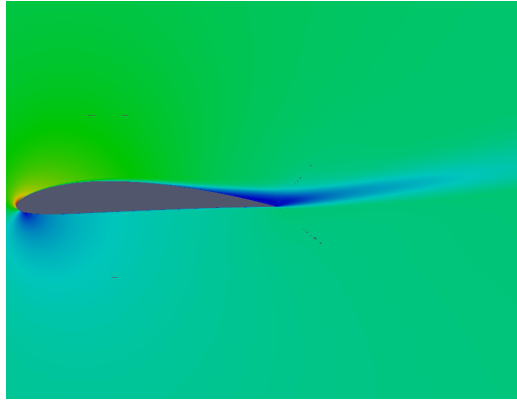


Fig. 5.19: Velocity magnitude contour plot of the NACA 4412 airfoil

Figure 5.19 shows a contour plot of the velocity magnitude over the airfoil. Good wake resolution can be seen. The surface pressure coefficient for $\alpha = 13.87^\circ$ is shown in Figure 5.20, and compared with the experimental data of Coles and Wadcock [49], and the CFD code CFL3D. Excellent agreement can be seen between the strand-Cartesian solver, CFL3D and the experimental data of Coles and Wadcock.

Velocity profiles at six different locations along the trailing edge of the upper surface are shown in Figure 5.21. Relatively good agreement is found between CFL3D and the strand-Cartesian solver. The velocity profiles at $x = 0.9528$ and $x = 0.8973$ show reverse flow close to surface of the airfoil in both the x and y directions, which is indicative of the flow separation, possibly the formation of a laminar separation bubble. More accurate results may possibly be obtained if preconditioning were to be implemented. The trailing

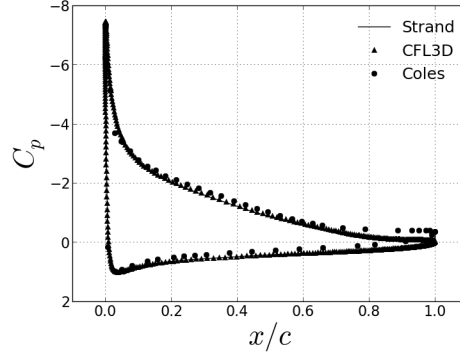


Fig. 5.20: Pressure coefficient for flow over a NACA 4412 airfoil at $M = 0.09$ and $Re = 1.52 \times 10^6$ at an angle of attack of 13.87° .

edge of the airfoil encounters very low speed due to the separation of flow. Preconditioning would help by accelerating the convergence to a steady state for problems where a significant portion of the flow is low speed.

Table 5.4 shows the coefficient of lift and drag for CFL3D, FUN3D and the strand-Cartesian solver. The strand-Cartesian solver predicted a coefficient of lift close to FUN3D and CFL3D. The drag prediction showed some discrepancy with FUN3D and CFL3D. It is possible to attribute this inaccuracy to the lack of preconditioning in the solver.

Figure 5.22 shows a direction field plot of the trailing edge. The arrows have been made a uniform size, rather than sized by the velocity magnitude, in order to make visualization easier. Reverse flow in the plot is clearly evident, and prediction of the formation of a laminar separation bubble on the trailing edge is confirmed.

Table 5.4: Comparison of computed lift and drag coefficients for flow over a NACA 4412 airfoil. at $M = 0.09$, $\alpha = 13.87^\circ$ $Re = 1.52 \times 10^6$.

	C_l	C_d
Strand (513 points)	1.69013	0.031477
FUN3D (513 points)	1.7210	0.02947
CFL3D (513 points)	1.7170	0.02861

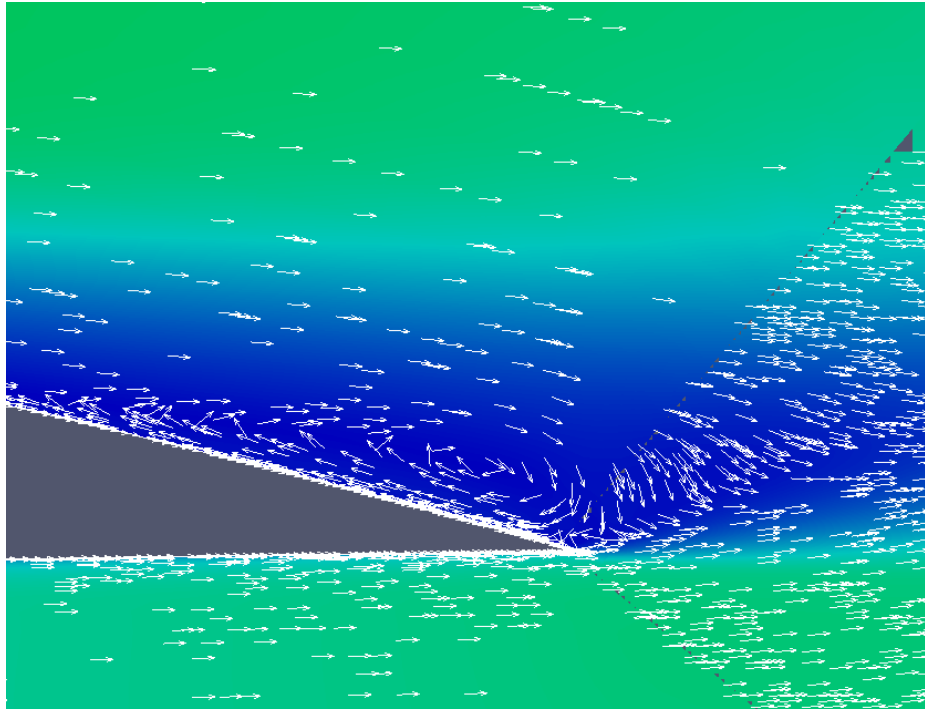
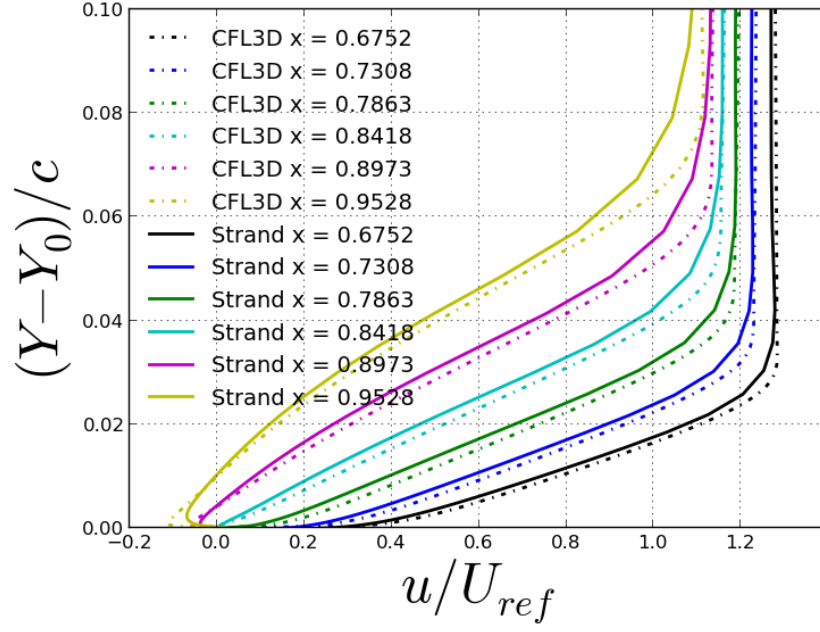
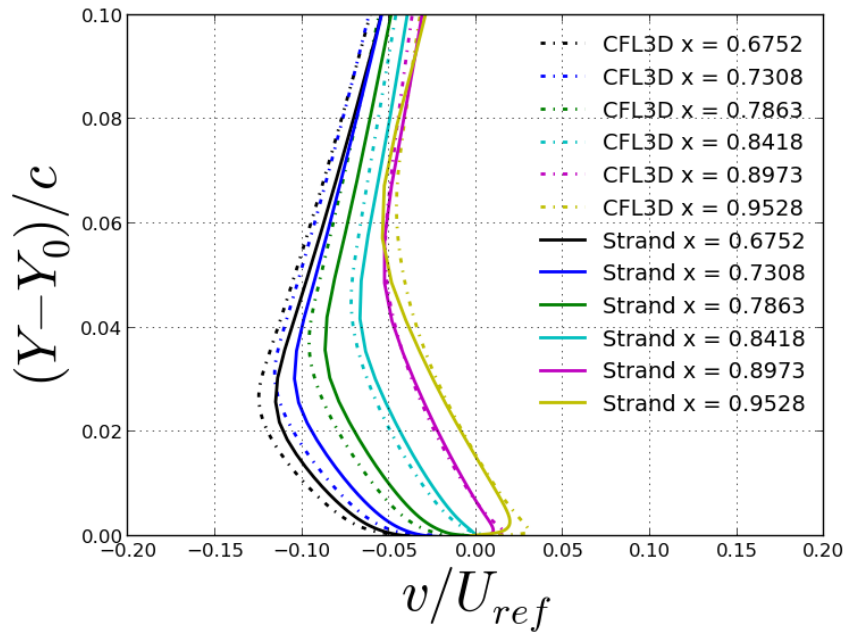


Fig. 5.22: Direction field plot of the trailing edge separation over the NACA 4412 airfoil



(a) U Velocity



(b) V Velocity

Fig. 5.21: Velocity profiles along the NACA 4412 trailing edge

Chapter 6

Conclusions

6.1 Conclusions

The strand-Cartesian grid method is examined for its ability to compute high Reynolds number flows over a variety of geometry using the Spalart-Allmaras (SA) turbulence model. Verification of the model is undertaken to ensure correct implementation. The Method of Manufactured Solutions grid convergence verification method showed the desired second order convergence was achieved for all system equations.

The validation procedure was carried out by comparing results from the strand-Cartesian solver to two independent compressible codes; FUN3D and CFL3D. The zero pressure gradient flat plate was run at $Re = 5 \times 10^6$. The strand solver computed drag to within 0.7% and 1.5% of a finer-grid FUN3D and CFL3D case respectively. Strand grid turbulent viscosity and velocity profiles at various locations along the plate were shown to agree well with the FUN3D and CFL3D.

The bump-in-channel case explored $Re = 3 \times 10^6$ flow through a channel with a pressure gradient caused by a bump. The strand solver computed drag to within 0.3% and 0.09% of a finer-grid FUN3D and CFL3D case respectively. Strand grid turbulent viscosity and velocity profiles at various locations on the bump were shown to agree excellently with the FUN3D and CFL3D. Surface pressure and skin friction coefficient along the bump was also shown to agree excellently with the FUN3D and CFL3D.

NACA 0012 case examined several candidate meshing strategies, including strand smoothing and multi-strands, with and without telescoping Cartesian refinement into sharp corner regions. For turbulent $Re = 5 \times 10^6$ flow over a NACA 0012 airfoil, smooth strands with local Cartesian refinement around the sharp trailing edge computed lift to within 0.1%

of a finer-grid FUN3D case, and drag to within 1.3%. The best multi-strand result was a difference of 0.5% in lift and 5.7% in drag.

NACA 4412 case examines strand-Cartesian’s ability to predict flow separation on the trailing edge of an airfoil. For turbulent $Re = 1.52 \times 10^6$ flow with a Mach number of 0.09 over a NACA 4412 airfoil, smooth strands with local Cartesian refinement around the sharp trailing edge computed lift to within 1.79% and drag to within 6.8% for a FUN3D case, and lift to within 1.56% and drag to within 10% for a CFL3D case. Velocity profiles show reasonable agreement with CFL3D, however implementing preconditioning to the solver in the future may increase the accuracy of the solution.

6.2 Future Work

The Spalart-Allmaras model has been shown to excel in the strand-Cartesian solver with the given cases studies. Further cases such as a “backwards facing step” should be explored. However, cases such as these provide a challenge in meshing with strand grids. Three-dimensional cases studies should be performed, however the challenges with strand meshing in 3D must be addressed first. Both internal and external sharp corner treatments must be examined further. For example, a “3D Supersonic Square Duct” validation case requires a highly resolved boundary layer on adjacent walls of a square duct, causing mesh generation issues at the internal corner. Such challenges need be overcome before a strand-Cartesian solver may be validated for such a case.

The SA model, while excellent with external flows, has been found to perform poorly (or at least worse than other models) with interior flows; such as pipe flow [20]. It would be appropriate then, to implement a turbulence model that excels in scenarios that SA model struggles with. One such model is the $k - \omega$ Menter SST model. Like the SA model, this is a common turbulence model with plenty of documentation, and its own advantages and disadvantages. Unlike the SA model, the $k - \omega$ Menter SST model has been shown to excel in interior flows [51]. Verification and validation should be undertaken in similar fashion to this work.

Alternatively, Large-Eddy Simulation (LES) could be implemented into the strand-Cartesian solver to provide highly accurate turbulence simulations. An LES strand-Cartesian solver could prove to be extremely powerful. Other work that could be explored includes developing high-order methods for strands, or implementing automated grid adaptation methods.

References

- [1] Katz, A., Wissink, A., Sankaran, V., Meakin, R., and Chan, W., “Application of Strand Meshes to Complex Aerodynamic Flow Fields,” *Journal of Computational Physics*, Vol. 230, 2011, pp. 6512–6530.
- [2] Reynolds, O., “An Exprimantal Investigation of the Circumstances which determine whether the Motion of Water Shall be Direct or Sinous, and the Law of Resistance in Parallel Channels,” Tech. Rep., 1883.
- [3] Rumsey, C., “NASA Langley Turbulence Modeling Resource (<http://turbmodels.larc.nasa.gov>),” 2012.
- [4] Meakin, R., Wissink, A., Chan, W., Pandya, S., and Sitaraman, J., “On Strand Grids for Complex Flows,” *AIAA Paper* 2007-3834, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.
- [5] Wissink, A., Potsdam, M., Sankaran, V., Sitaraman, J., Yang, Z., and Mavriplis, D., “A Coupled Unstructured-Adaptive Cartesian CFD Approach for Hover Prediction,” Tech. rep., American Helicopter Society 66th Annual Forum, Phoenix, AZ, May 2010.
- [6] Wissink, A., Katz, A., Chan, W., and Meakin, R., “Validation of the Strand Grid Approach,” *AIAA Paper* 2009-3792, AIAA 19th Computational Fluid Dynamics Conference, San Antonio, TX, June 2009.
- [7] Steger, J., Dougherty, F., and Benek, J., “A Chimera Grid Scheme,” Tech. rep., ASME Mini-Symposium on Advances in Grid Generation, Houston, TX, June 1983.
- [8] Benek, J. A., Steger, J. L., and Dougherty, F. C., “A Flexible Grid Embedding Technique with Application to the Euler Equations,” *AIAA paper* 1983-1944, AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, July 1983.
- [9] Lee, Y-L., and Baeder, J., “Implicit Hole Cutting – A New Approach to Overset Grid Connectivity,” *AIAA Paper* 2003-4128, AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, June 2003.
- [10] Sitaraman, J., Floros, M., Wissink, A., and Potsdam, M., “Parallel Domain Connectivity Algorithm for Unsteady Flow Computations Using Overlapping and Adaptive Grids,” *Journal of Computational Physics*, Vol. 229, 2008, pp. 4703–4723.
- [11] Feynman, R., *The Feynman Lectures on Physics: Mainly Mechanics, Radiation and Heat*, 6th ed., Basic Books, New York, NY, 1977.
- [12] Work, D., Tong, O., Workman, R., Katz, A., and Wissink, A., “Strand Grid Solution Procedures for Sharp Corners,” Tech. Rep., AIAA 52nd Aerospace Sciences Meeting, Dallas, TX, January 2013.

- [13] Cengel, Y., and Cimbala, J., *Fluid Mechanics: Fundamentals and Applications*, 2nd ed., McGraw Hill, New York, NY, 2010.
- [14] Kundu, P., and Cohen, I., *Fluid Mechanics*, 4th ed., Academic Press, Burlington, MA, 2008.
- [15] Kolmogorov, A., “The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds Numbers,” Tech. Rep., Proceedings of the Royal Society, July 1991.
- [16] Moin, P., and Mahesh, K., “Direct Numerical Simulation: A Tool in Turbulence Research,” Tech. Rep., 1998.
- [17] Meneveau, C., and Katz, J., “Scale-Invariance and Turbulence Models for Large-Eddy Simulation,” Tech. Rep., January 2000.
- [18] Wilcox, D., *CFD Modeling for CFD*, 2nd ed., DCW Industries, Glendale, CA, 1994.
- [19] Spalart, P., and Allmaras, S., “A One-Equation Turbulence Model for Aerodynamic Flows,” *Recherche Aerospaciale*, Vol. 1, 1994, pp. 5–21.
- [20] Javaherchi, T., “Review of the Spalart-Allmaras Turbulence Model and its Modifications,” Tech. Rep., 2010.
- [21] AIAA, “Guide for the Verification and Validation of Computational Fluid Dynamics Simulations,” Tech. Rep., AIAA Guide, 1998.
- [22] Roache, P., “Code Verification by the Method of Manufactured Solutions,” *Transactions of the ASME*, Vol. 124, 2002, pp. 4–10.
- [23] Roy, C., Nelson, C., Smith, T., and Ober, C., “Verification of Euler/Navier-Stokes Codes Using the Method of Manufactured Solutions,” *International Journal for Numerical Methods in Fluids*, Vol. 44, 2004, pp. 599–620.
- [24] Veluri, S., Roy, C., and Luke, E., “Comprehensive Code Verification for an Unstructured Finite Volume CFD Code,” AIAA Paper 2010-127, AIAA 48th Aerospace Sciences Meeting, Orlando, FL, January 2010.
- [25] Diskin, B., and Thomas, J., “Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Inviscid Fluxes,” AIAA Paper 2010-1079, AIAA 48th Aerospace Sciences Meeting, Orlando, FL, January 2010.
- [26] Diskin, B., Thomas, J., Nielsen, E., Nishikawa, H., and White, J., “Comparison of node-centered and cell-centered unstructured finite-volume discretizations. Part I: viscous fluxes,” AIAA Paper 2009-0597, 2009.
- [27] Diskin, B., and Thomas, J., “Effects of Mesh Regularity on Accuracy of Finite-Volume Schemes,” AIAA Paper 2012-0609, AIAA 50th Aerospace Sciences Meeting, Nashville, TN, January 2012.

- [28] Eriksson, S., and Nordström, J., “Analysis of mesh and boundary effects on the accuracy of node-centered finite volume schemes,” AIAA Paper 2009-3651, AIAA 19th Computational Fluid Dynamics Conference, San Antonio, TX, June 2009.
- [29] Katz, A., and Sankaran, V., “Mesh Quality Effects on the Accuracy of Euler and Navier-Stokes Solutions on Unstructured Meshes,” Tech. Rep., 6th International Conference on Computational Fluid Dynamics, St. Petersburg, Russia, July 2010.
- [30] Diskin, B., and Thomas, J., “Accuracy Analysis for Mixed-Element Finite-Volume Discretization Schemes,” NIA Report 2007-08, National Institute of Aerospace, 2007.
- [31] Roy, C., “Review of Code and Solution Verification Procedures for Computational Simulation,” *Journal of Computational Physics*, Vol. 205, 2005, pp. 131–156.
- [32] Luke, E., Hebert, S., and Thompson, D., “Theoretical and Practical Evaluation of Solver-Specific Mesh Quality,” AIAA paper 2008-0934, AIAA 46th Aerospace Sciences Meeting, Reno, NV, January 2008.
- [33] Giles, M., “Accuracy of Node-based Solutions on Irregular Meshes,” *Lecture Notes in Physics*, Vol. 323, 1989, pp. 273–277.
- [34] Folkner, D., Katz, A., and Sankaran, V., “Design and Verification Methodology of Boundary Conditions for Finite Volume Schemes,” Tech. rep., 7th International Conference on Computational Fluid Dynamics, Big Island of Hawaii, July 2012.
- [35] Katz, A., “Course Notes for MAE 6440 - Advanced Computational Fluid Dynamics,” Tech. Rep., Utah State University, Logan, UT, 2012.
- [36] Oliver, T., “Farve-Averaged Navier-Stokes and Turbulence Model Equation Documentation,” Tech. Rep., 2009.
- [37] Wilcox, D., “Reassessment of the Scale-Determining Equation for Advanced Turbulence Models,” *AIAA Journal*, Vol. 26, No. 11, Novemebr 1988, pp. 1299–1310.
- [38] Schmitt, F., “About Boussinesq’s Turbulent Viscosity Hypothesis: Historical Remarks and a Direct Evaluation of Its Validity,” *Comptes Rendus Mecanique Journal*, 2007, pp. 617–627.
- [39] Catris, S., and Aupoix, B., “Density Corrections for Turbulence Models,” *Aerospace Science and Technology*, Vol. 4, 2000, pp. 1–11.
- [40] Allmaras, S., Johnson, F., and Spalart, P., “Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model,” Tech. rep., 7th International Conference on Computational Fluid Dynamics, Big Island of Hawaii, July 2012.
- [41] Oliver, T., *A High-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier-Stokes Equations*, Ph.D. thesis, Massachusetts Institute of Technology, 2008.
- [42] Gustafson, K., *Introduction to Partial Diferential Equations and Hilbert Space Methods*, 3rd ed., Dover, Mineola, 1999.

- [43] Allmaras, S., “Lagrange Multiplier Implementation of Dirichlet Boundary Conditions in Compressible Navier-Stokes Finite Element Methods,” AIAA paper 2005-4714, AIAA 17th Computational Fluid Dynamics Conference, Toronto, June 2005.
- [44] Katz, A., and Sankaran, V., “Mesh Quality Effects on the Accuracy of Euler and Navier-Stokes Solutions on Unstructured Meshes,” *Journal of Computational Physics*, Vol. 230, No. 20, 2011, pp. 7670–7686.
- [45] Swanson, R., and Turkel, E., “Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations,” AIAA paper 1987-1107, AIAA 8th Computational Fluid Dynamics Conference, Honolulu, HI, June 1987.
- [46] Gregory, N., and O’Reilly, C., “Low-Speed Aerodynamic Characteristics of NACA 0012 Aerofoil Sections, including the Effects of Upper-Surface Roughness Simulation Hoar Frost,” *NASA RM 3726*, January 1970.
- [47] Ladson, C., “Effects of Independent Variation of Mach and Reynolds Numbers on the Low-Speed Aerodynamic Characteristics of the NACA 0012 Airfoil Section,” *NASA TM 4074*, October 1988.
- [48] Sankaran, V., Sitaraman, J., Wissink, A., Datta, A., Jayaraman, B., Potsdam, M., Mavriplis, D., Yang, Z., O’Brien, D., Saberi, H., Cheng, R., Hariharan, N., and Strawn, R., “Application of the Helios Computational Platform to Rotorcraft Flowfields,” *AIAA Paper 2010-1230*, AIAA 48th Aerospace Sciences Meeting, Orlando, FL, January 2010.
- [49] Coles, D., and Wadcock, A. J., “Flying-Hot-Wire Study of Flow Past an NACA 4412 Airfoil at Maximum Lift,” *AIAA Journal*, Vol. 17, No. 4, April 1979, pp. 321–329.
- [50] Wadcock, A. J., “Structure of the Turbulent Separated Flow Around a Stalled Airfoil,” Tech. rep., NASA CR-152263, February 1979.
- [51] Menter, F. R., “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Application,” *AIAA Journal*, Vol. 32, No. 8, August 1994, pp. 1598–1605.